Micro Fiche Scan

Name of device(s) tested:
    TU80
Test description:
    TU80 FRONT END DIAG
MAINDEC Number or Package Identifier (after SEP 1977):
    CZTUWA0
Fiche Document Part Number:
    AH-T328A-MC
Fiche preparation date unknown, using copyright year:
    1983
Image resolution:
    1-bit black&white, compressed for minimal file size

.REM\

IDENTIFICATION

PRODUCT ID: AC-T327A-MC

PRODUCT TITLE: CZTUWA0 TU80 FRONT END DIAG

PRODUCT DATE: 24 - APRIL - 1983

MAINTAINER: TAPE DIAGNOSTIC ENGINEERING

AUTHOR: DICE SYSTEMS, INC.

TABLE OF CONTENTS

ABSTRACT

1.0 ABSTRACT

THIS IS A PDP-11/LSI RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TU80 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A PDP-11 SYSTEM. THE PROGRAM PROVIDES ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS THAT AID IN THE REPAIR OF THE DEVICE. REFERENCE THE FOLLOWING DIGITAL EQUIPMENT DOCUMENTS:

1. ENGINEERING SPECIFICATION FOR TU80 MAGTAPE CONTROLLER; DOCUMENT NUMBER: YM-C194D-022; REVISION NUMBER 2; DATE: 28-JUL-81.

2. ENGINEERING SPECIFICATION FOR TU80 DIAGNOSTIC PACKAGE; DOCUMENT NUMBER: YM-C194F-00; REVISION NUMBER 0; DATE: 2-SEP-81.

3. ENGINEERING SPECIFICATION FOR TU80 MAGTAPE SUBSYSTEM; DOCUMENT NUMBER: YM-C194S-02; REVISION NUMBER 3; DATE: 10-JUN-81.

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

HARDWARE, SOFTWARE REQUIREMENTS AND PREREQUISITES

2.0 HARDWARE, SOFTWARE REQUIREMENTS AND PREREQUISITES

2.1 HARDWARE REQUIREMENTS

PDP-11 FAMILY PROCESSOR WITH 32K WORDS OF MEMORY
TU80 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE I.E. 4K FOR I/O PAGE)

2.2 OPTIONAL HARDWARE:

UP TO 8 TU80 CONTROLLERS PER PDP-11 UP TO 1 DRIVES PER CONTROLLER

2.3 SOFTWARE REQUIREMENTS

PDP-11 DIAGNOSTIC SUPERVISOR (HSAADO.SYS)
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

2.4 PREREQUISITES

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

OPERATING INSTRUCTIONS-OPERATOR COMMANDS

3.0 OPERATING INSTRUCTIONS-OPERATOR COMMANDS

3.1 OPERATOR COMMANDS

THE TU80 DIAGNOSTIC IS A PDP-11 DIAGNOSTIC SUPERVISOR COMPATIBLE PROGRAM. 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)

```
CHMDLB0 XXDP+ DL MONITOR 28K
BOOTED VIA UNIT 0

ENTER DATE (DD-MMM-YY): 29-JAN-82
RESTART ADDRESS: 153726
50 HZ? N <CR>


LSI? N Y
THIS IS XXDP+ TYPE "H" OR "H/L" FOR DETAILS

R CZTUWA

CZTUWABINDRS LOADED
DIAG. RUN TIME SERVICES REV D. APR 79
CZTUW-A-0
****TU80 LOGIC DIAGNOSTIC****
UNIT IS TU80
DR>
DRS>START/FLAG:PNT:HOE
```

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) RESPONSE TO THE "CHANGE HW?" QUESTION, THE DIAGNOSTIC WILL NOT RUN. IT WILL GIVE THE MESSAGE "NO UNIT". A "Y" (YES) IS REQUIRED, AND AT LEAST A "1" IS REQUIRED AT

THE "#UNITS (D)?" QUESTION.

TSBA/TSDB=172520, VECTOR =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 M7454 CONTROLLERS
                PRESENT TO BE TESTED>


UNIT 0

DEVICE ADDRESS (O) 172520 ? <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.>
```

OPERATING INSTRUCTIONS - SAMPLE PRINTOUTS

4.0 OPERATING INSTRUCTIONS - SAMPLE PRINTOUTS

4.1 EXAMPLE OF ALL TESTS RUN TOGETHER

```
TST: 001 INITIALIZE #1
TST: 002 RAM TEST
TST: 003 COMMAND REJECT TEST
TST: 004 WRITE CHARACTERISTICS TEST
TST: 005 VOLUME CHECK
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

0 ERRORS

NOTE: PROGRAM NOW STARTS OVER AGAIN AT TEST 1
```

OPERATING INSTRUCTIONS - SAMPLE ERROR MESSAGES

5.0 OPERATING INSTRUCTIONS - SAMPLE ERROR MESSAGES

ERROR MESSAGE EXAMPLE 1

```
TST: 001 FIFO EXERCISER TEST
CZTUA HRD ERR 01610 ON UNIT 00 TST 016 SUB 000 PC: 040624
FIFO STATUS (IN WORD  9) INCORRECT AFTER WRITE FIFO

 TAPE BUS SIGNALS IN WORD #8: - DESIGNATOR <BIT #>
PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>
IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>
IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>
 TAPE BUS SIGINALS IN WORD #9:
DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>
 MESSAGE BUFFER ADDRESS = 047352
 MESSAGE BUFFER CONTENTS:
WORD #0 EXPD: 100020  RECV: 100020  XOR: 000000
WORD #1 EXPD: 000012  RECV: 000012  XOR: 000000
WORD #2 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #3 EXPD: 000010  RECV: 000010  XOR: 000000
WORD #4 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #5 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #6 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #7 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #8 EXPD: 070217  RECV: 070217  XOR: 000000
WORD #9 EXPD: 000074  RECV: 000034  XOR: 000040
```

ERROR MESSAGE EXAMPLE 2

```
CZTUWA HRD ERR 00159 ON UNIT 00 TST 001 SUB 005 PC: 026202
TSSR NOT CORRECT AFTER SPACE RECORDS COMMAND
 TSSR = 100214
 TSSR BITS SET: SC,SSR
 TERMINATION CLASS CODE = UNRECOVERABLE ERROR
 PACKET ADDRESS = 026420
 PACKET WORD #  = 140010
 PACKET WORD #  = 000010
 PACKET WORD #  = 000000
 PACKET WORD #  = 000024
```

ERROR MESSAGE EXAMPLE 3

```
CZTUWA HRD ERR 00121 ON UNIT 00 TST 001 SUB 002 PC: 023306
MOT BIT (XSTO) NOT SET DURING REWIND (EXTENDED FEATURES MODE)
EXPD: 000312 RECV: 000112 XOR: 000200
```

PROGRAM RUN TIMES

6.0 PROGRAM RUN TIMES

THE AVERAGE RUN TIMES OF THE PROGRAM 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-34 CONSOLE.

THE PROGRAM RUNS IN TWO MODES; NO ITERATIONS AND DEFAULT MODE. IN THE NO ITERATIONS MODE, EACH TEST IS RUN ONCE, WITH NO ITERATIONS. IN THE DEFAULT MODE EACH TEST IS REPEATED BY THE NUMBER OF TIMES INDICATED BY THE ITERATION COUNT. NO ITERATIONS MODE IS SELECTED BY ANSWERING THE INHIBIT ITERATIONS QUESTION WITH A "Y" (YES).

TEST NUMBER	N/I SECS.	ITER SECS	DEF SECS.
1	1	30	29
2	25	120	95
3	25	475	450
4	20	20	0
5	1	10	9
6	20	20	0
7	2	1	1
8	8	11	3
9	1	1	0
10	1	1	0
11	1	1	0

THE TIMES REQUIRED TO RUN TESTS 1 THROUGH 12 IN ONE COMMAND:

Q.V.	1 MIN 57 SECONDS
DEFAULT	12 MINS

7.0 TEST DESCRIPTIONS

7.1 TEST 1 - INITIALIZATION TEST #1

```
*************************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S                *
* CONTROLLER (M7454)                                                    *
*************************************************************************
```

THIS TEST VERIFIES THAT THE M7454 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 IGNORE 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. SUBTEST RUN FORCES FORCES SELF-TEST TO RUN BY DOING A BYTE WRITE TO THE TSSR.

7.2 TEST 2 - RAM TEST

```
*************************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S                *
* CONTROLLER (M7454)                                                    *
*************************************************************************
```

THIS TEST VERIFIES THAT ALL LOCATIONS OF THE RAM ON THE M7454 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.

7.2.1 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. THE LOW BYTE OF TSDB.

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.

7.2.2 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.

7.2.3 TEST 2, SUBTEST 3:-

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

7.3 TEST 3: COMMAND REJECT

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S             *
* CONTROLLER (M7454)                                                 *
**********************************************************************
```

THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE CHARACTERISTICA 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 MAINTANENCE MODE WRAP AROUND FUNCTIONS, THE COMMAND IMAGE BLOCK IN THE M7454'S RAM (LOCATIONS 20 - 27 (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 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.

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.:

7.4 TEST 4 - WRITE CHARACTERISTICS

```
**************************************************************************
* NOTE: IF THIS TEST DETECTS A FAILURE REPLACE THE TU80'S                *
* CONTROLLER (M7454)                                                     *
**************************************************************************
```

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 INDICATE A BAD M7454 MODULE.

7.4.1 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 M7454 RAM ARE CORRECT.

7.4.2 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.

7.4.3 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

7.5 TEST 5: VOLUME CHECK

****
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S *
* CONTROLLER (M7454) *
****

THIS TEST VERIFIES THAT THE VOLUME CHECK (VCK) BIT, A FLAG HELD WITHIN THE M7454 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 PREVOUS 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). WHICH 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 SHOULD CONTAIN ALL 1'S.

4. 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.

7.6 TEST 6 - COMPLETION INTERRUPT

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S             *
* CONTROLLER (M7454)                                                 *
**********************************************************************
```

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 XSTO OF ANY MESSAGE PACKET IS SET AND THAT A COMPLETTION 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.

7.7 TEST 7 - BASIC PACKET PROTOCAL

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S             *
* CONTROLLER (M7454)                                                 *
**********************************************************************
```

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.

7.7.1 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.

7.7.2 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.

7.7.3 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.

7.7.4 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 THAT THE FIRST COMMAND COMPLETED PROPERLY, THAT RMR IS SET, AND THAT THE SECOND COMMAND IS IGNORED.;

7.8 TEST 8 - NON-TAPE MOTION COMMANDS

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S             *
* CONTROLLER (M7454)                                                 *
**********************************************************************
```

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

TEST 9: 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 XST0 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 XST0 IS 0.

7.9 TEST 9 - MEMORY ADDRESSING TEST

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S             *
* CONTROLLER (M7454)                                                 *
**********************************************************************
```

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.

7.9.1. TEST 9, SUBTEST 1:-

THIS SUBTEST VERIFIES THE CONTROLLER CAN FETCH A GET STATUS COMMAND FROM ALL AVILABLE 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.

7.9.2 TEST 9, SUBTEST 2:-

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 FIRST A 1 THEN A 0 THROUGH THE ADDRESS BITS.

7.9.3 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.

7.9.4 TEST 9, SUBTEST 4:-

THIS SUBTEST VERIFIES 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).

7.10 TEST 10 - INITIALIZE #3

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S             *
* CONTROLLER (M7454)                                                 *
**********************************************************************
```

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

7.11 TEST 11 - BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S             *
* CONTROLLER (M7454)                                                 *
**********************************************************************
```

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 BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT,BGNSETUP
002000                                  HEADER  CZTUW,A,0,655.,0
002000                          LSNAME::                ;DIAGNOSTIC NAME
002000  103                             .ASCII /C/
002001  132                             .ASCII /Z/
002002  124                             .ASCII /T/
002003  125                             .ASCII /U/
002004  127                             .ASCII /W/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010                          LSREV::                 ;REVISION LEVEL
002010  101                             .ASCII  /A/
002011                          LSDEPO::                ;0
002011  060                             .ASCII  /0/
002012                          LSUNIT::                ;NUMBER OF UNITS
002012  000001                          .WORD   T$PTHV
002014                          LSTIML::                ;LONGEST TEST TIME
002014  001217                          .WORD   655.
002016                          LSHPCP::                ;POINTER TO H.W. QUES.
002016  052640                          .WORD   L$HARD
002020                          LSSPCP::                ;POINTER TO S.W. QUES.
002020  053000                          .WORD   L$SOFT
002022                          LSHPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002124                          .WORD   L$HW
002024                          LSSPTP::                ;PTR. TO S.W. PTABLE
002024  002134                          .WORD   L$SW
002026                          LSLADP::                ;DIAG. END ADDRESS
002026  053220                          .WORD   L$LAST
002030                          LSSTA::                 ;RESERVED FOR APT STATS
002030  000000                          .WORD   0
002032                          LSCO::
002032  000000                          .WORD   0
002034                          LSDTYP::                ;DIAGNOSTIC TYPE
002034  000000                          .WORD   0
002036                          LSAPT::                 ;APT EXPANSION
002036  000000                          .WORD   0
002040                          LSDTP::                 ;PTR. TO DISPATCH TABLE
002040  053166                          .WORD   L$DISPATCH
002042                          LSPRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000000                          .WORD   0
```

PROGRAM HEADER

```
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  022702                          .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  022400                          .WORD   L$AU
002072                          L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  022476                          .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  021602                          .WORD   L$INIT
002106                          L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  022660                          .WORD   L$CLEAN
002110                          L$ACP::                 ;PTR. TO AUTO CODE
002110  022604                          .WORD   L$AUTO
002112                          L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021572                          .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:
```

```
                                .SBTTL  GLOBAL EQUATES SECTION


                                .SBTTL  GLOBAL EQUATES SECTION

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

                                EQUALS          ; GET STANDARD EQUATES.
                        ;
                        ; BIT DIFINITIONS
                        ;
100000                  BIT15== 100000
040000                  BIT14== 40000
020000                  BIT13== 20000
01C000                  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
    949
    950 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
```

```
                                .SBTTL  TU80 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)
                        ;-
```

```
1012          100000                  X1.DLT  = BIT15                         ;DATA LATE
1013          040000                  X1.SPARE= BIT14                         ;NOT USED
1014          020000                  X1.COR  = BIT13                         ;CORRECTABLE DATA ERROR
1015          017375                  X1.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT0 ;ALWAYS 0
1016          000400                  X1.RBP  = BIT8                          ;READ BUS PARITY ERROR
1017          000002                  X1.UNC  = BIT1                          ;UNCORRECTABLE DATA OR HARD ERROR
1018
1019                                  ;+
1020                                  ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
1021                                  ;(XST2)
1022                                  ;-
1023          100000                  X2.OPM  = BIT15                         ;OPERATION IN PROGRESS (TAPE MOVING)
1024          040000                  X2.RCE  = BIT14                         ;RAM CHECKSUM ERROR
1025          035400                  X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8   ;NOT USED BY TU80 (ALWAYS=0)
1026          002000                  X2.WCF  = BIT10                         ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
1027          000200                  X2.EXTF = BIT7                          ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
1028          000100                  X2.BUFE = BIT6                          ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
1029          000077                  X2.REV  = 000077                        ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
1030          000007                  X2.UNIT = BIT2+BIT1+BIT0                ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
1031
1032                                  ;+
1033                                  ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
1034                                  ;(XST3)
1035                                  ;-
1036          177400                  X3.MDE  = 177400                        ;MICRO-DIAGNOSTIC ERROR CODE
1037          000200                  X3.SPARE= BIT7                          ;NOT USED BY TU80
1038          000100                  X3.OPI  = BIT6                          ;OPERATION INCOMPLETE
1039          000040                  X3.REV  = BIT5                          ;REVERSE
1040          000020                  X3.TRF  = BIT4                          ;TRANSPORT RESPONSE FAILURE
1041          000010                  X3.DCK  = BIT3                          ;DENSITY CHECK
1042          000006                  X3.MBZ  =BIT2+BIT1                      ;NOT USED ALWAYS 0
1043          000001                  X3.RIB  = BIT0                          ;REVERSE INTO BOT
1044
1045                                  ;+
1046                                  ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
1047                                  ;(XST4)
1048                                  ;-
1049          100000                  X4.HSP  = BIT15                         ;HIGH SPEED
1050          040000                  X4.RCE  = BIT14                         ;RETRY COUNT EXCEEDED
1051          020000                  X4.TSM  = BIT13                         ;TRANSPORT SPECIAL MODE
1052          017400                  X4.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT8   ;NOT USED ALWAYS 0
1053          000377                  X4.WRC  = 000377                        ;WRITE RETRY COUNT FIELD
1054
1055
1056                                  ;+
1057                                  ;
1058                                  ;TSSR TERMINATION CODES (BIT 0-2)
1059                                  ;
1060                                  ;-
1061
1062          000006                          TSREJ=  3+2                     ;COMMAND REJECTED
1063          000006                          UNREC=  6                       ;UNRECOVERABLE ERROR
1064
1065                                  ;+
1066                                  ;
1067                                  ;DEVICE REGISTER OFFSETS
1068                                  ;
```

```
                                ;-
                                
            177776              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
```

```
1126    000117                  RMMSGEND= 117                           ;MESSAGE BUFFER END RAM ADDRESS
1127                            ;*
1128                            ;
1129                            ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
1130                            ;
1131                            ;-
1132
1133    000006                  XST0== 6                                ;EXTENDED STATUS REGISTER 0 (WORD 4)
1134    000010                  XST1== 8.                               ;EXTENDED STATUS REGISTER 1 (WORD 5)
1135    000012                  XST2== 10.                              ;EXTENDED STATUS REGISTER 2 (WORD 6)
1136    000014                  XST3== 12.                              ;EXTENDED STATUS REGISTER 3 (WORD 7)
1137    000016                  XST4== 14.                              ;EXTENDED STATUS REGISTER 4 (WORD 8)
1138
1139
1140                            ;*
1141                            ;
1142                            ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
1143                            ;
1144                            ;-
1145
1146    000002                          PKLOW   = 2                     ;LOW ORDER CHARACTERISTIC DATA POINTER
1147    000004                          PKHI    = 4                     ;HIGH ORDER CHARACTERISTIC DATA POINTER
1148    000006                          PKBCNT  = 6                     ;NUMBER OF BYTES IN DATA PACKET
1149
1150    000010                          EXBCNT=10                       ;NUMBER OF BYTES IN EXTENDED DATA PACKET
1151
1152                            ;*
1153                            ;DATA PACKET OFFSETS FOR WRITE SUBSYTEM COMMAND
1154                            ;-
1155    000000                          BSEL0   = 0                     ;BYTE 0
1156    000001                          BSEL1   = 1                     ;BYTE 1
1157    000002                          SEL2    = 2                     ;WORD 2
1158    000004                          SELDATA = 4                     ;WORD 3
1159
1160                            ;*
1161                            ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
1162                            ;-
1163    000000                  PW.NOP          = 0                     ;NO-OP
1164    000001                  PW.RDRAM        = 1                     ;READ RAM
1165    000002                  PW.WTRAM        = 2                     ;WRITE RAM
1166    000003                  PW.RFIFO        = 3                     ;READ FIFO
1167    000004                  PW.WFIFO        = 4                     ;WRITE FIFO
1168    000005                  PW.RDSTAT       = 5                     ;READ STATUS
1169    000006                  PW.WCTL         = 6                     ;WRITE TAPE CONTROL
1170    000007                  PW.WFRT         = 7                     ;WRITE TAPE FORMAT
1171    000010                  PW.WMISC        = 10                    ;WRITE MISCELLANEOUS
1172    000011                  PW.WNPR         = 11                    ;WRITE NPR CONTROL
1173    000020                  PW.D22          = 20                    ;DO MICROTEST 22
1174    000021                  PW.D11          = 21                    ;DO MICROTEST 11
1175    000022                  PW.D13          = 22                    ;DO MICROTEST 13
1176    000023                  PW.MO1311       = 23                    ;DISABLE MICROTEST 11 AND 13
1177    000024                  PW.RDEXT        = 24                    ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPO
1178
1179                            ;*
1180                            ;BSEL1 CODES FOR WRITE TAPE CONTROL
1181                            ;-
1182    000200                  WC.IFAD         = BIT7                  ;IFAD - FORMATTER ADDRESS
```

```
1183    000100          WC.IOTAD        = BIT6                  ;ITAD0  - TRANSPORT ADDRESS BIT 0
1184    000040          WC.I1TAD        = BIT5                  ;ITAD1  - TRANSPORT ADDRESS BIT 1
1185    000020          WC.ISRESV       = BIT4                  ;IRESV5 - RESERVED #5
1186    000010          WC.IREW         = BIT3                  ;IREW   - REWIND
1187    000004          WC.IRWU         = BIT2                  ;IRWU   - REWIND AND UNLOAD
1188    000002          WC.IFEN         = BIT1                  ;IFEN   - FORMATTER ENABLE
1189    000001          WC.IGO          = BIT0                  ;GO
1190
1191                    ;+
1192                    ;BSEL1 CODES FOR WRITE FORMAT
1193                    ;-
1194    000200          WF.IHISP        = BIT7                  ;IHISP  - HIGH SPEED
1195    000100          WF.IWRT         = BIT6                  ;IWRT   - WRITE
1196    000040          WF.IREV         = BIT5                  ;IREV   - REVERSE
1197    000020          WF.IWFM         = BIT4                  ;IWFM   - WRITE FILE MARK
1198    000010          WF.IEDIT        = BIT3                  ;IEDIT  - EDIT
1199    000004          WF.IERASE       = BIT2                  ;IERASE - ERASE
1200    000002          WF.I3RESV       = BIT1                  ;IRESV3 - RESERVED #3
1201    000001          WF.I4RESV       = BIT0                  ;IRESV4 - RESERVED #4
1202
1203
1204                    ;+
1205                    ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
1206                    ;-
1207    000200          MS.EXT          = BIT7                  ;INVERT SENSE OF EXTENDED FEATURES SWITCH
1208    000020          MS.RSFIFO       = BIT4                  ;RESET FIFO AND INPUT PARITY ERRORR
1209    000010          MS.RSTAPE       = BIT3                  ;RESET TAPE STATUS IN 2 FLIP-FLOPS
1210    000006          MS.ATTN         = BIT2!BIT1             ;ATTENTION TRIGGER FIELD
1211    000001          MS.RSD          = BIT0                  ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
1212                    ;+
1213                    ; MS.ATTN SUBCODES
1214                    ;-
1215    000000                  MSA.NOP = 0*2                   ;NO-OP (NOTHING TRIGGERED)
1216    000002                  MSA.VOL = 1*2                   ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
1217    000004                  MSA.NRAM= 2*2                   ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
1218    000006                  MSA.FRAM= 3*2                   ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
1219                    ;+
1220                    ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
1221                    ;-
1222    000200          NP.IR           = BIT7                  ;INTERRUPT REQUEST (0-1 TRANSITION)
1223    000100          NP.OUT          = BIT6                  ;TAPE DATA DIRECTION OUT (0= IN)
1224    000040          NP.LOOP         = BIT5                  ;ENABLE TRANSPORT LOOPBACK
1225    000020          NP.WRP          = BIT4                  ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
1226                    ;+
1227                    ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
1228                    ;-
1229
1230    000200          S2.DIM          = BIT7                  ;WORD #9 BYTE 2 DATA IN MISS
1231    000100          S2.ILW          = BIT6                  ;                ILW H
1232    000040          S2.OUTRDY       = BIT5                  ;                OUT RDY H
1233    000020          S2.INRDY        = BIT4                  ;                IN RDY H
1234    000010          S2.ATIMR        = BIT3                  ;                TIMER A FLAG H
1235    000004          S2.BTIMR        = BIT2                  ;                TIMER B FLAG H
1236    000003          S2.UNDEF        = BIT1+BIT0             ;(UNDEFINED)
1237    100000          S1.PARIN        = BIT15                 ;WORD #8 BYTE 1 PARIN H
1238    040000          S1.I2RESV       = BIT14                 ;                IRESV2
1239    020000          S1.I1RESV       = BIT13                 ;                IRESV1
```

```
1240    010000          S1.IEOT         = BIT12                 ;                       IEOT L
1241    004000          S1.IIDENT       = BIT11                 ;                       IIDENT H
1242    002000          S1.ICER         = BIT10                 ;                       ICER H
1243    001000          S1.IFMK         = BIT9                  ;                       IFMK H
1244    000400          S1.IHER         = BIT8                  ;                       IHER H
1245    000200          S0.ISPEED       = BIT7                  ;WORD #8 BYTE 0 ISPEED H
1246    000100          S0.IRDY         = BIT6                  ;                       IRDY L
1247    000040          S0.IONL         = BIT5                  ;                       IONL L
1248    000020          S0.ILDP         = BIT4                  ;                       ILDP L
1249    000010          S0.IDBY         = BIT3                  ;                       IDBY L
1250    000004          S0.IRWD         = BIT2                  ;                       IRWD L
1251    000002          S0.IFBY         = BIT1                  ;                       IFBY L
1252    000001          S0.IFPT         = BIT0                  ;                       IFPT L
1253                    ;       SPECIAL KEYBORD STUFF FOR MOVER PROGRAM
1254
1255    177560                  TKS     =177560                 ;KEYBOARD STATUS REGISTER
1256    177562                  TKB     =177562                 ;KEYBOARD DATA REGISTER
1257    177564                  TPS     =177564                 ;CONSOLE PRINTER STATUS REGISTER
1258    177566                  TPB     =177566                 ;CONSOLE PRINTER DATA REGISTER
1259    007776                  HIMEM   =007776                 ;HIGH MEMORY MASK VALUE
1260                    ;       CONTROLLER DEFINITIONS
1261    174400                  CSR     =174400                 ;STATUS AND CONTROL REGISTER
1262    174402                  BAR     =174402                 ;DL ADDRESS REGISTER
1263    174404                  DAR     =174404                 ;PLATTER ADDRESS
1264    174406                  MPR     =174406                 ;MULTIPURPOSE REGISTER
1265                    ;
1266                    ;       CONTROLLER COMMANDS
1267                    ;
1268    000004                  DLGETS  =4                      ;GET STATUS COMMAND
1269    000006                  SEEK    =6                      ;SEEK TRACK AND HEAD SELECT
1270    000010                  DLRDHD  =10                     ;READ SECTOR HEADER
1271    000014                  READ    =14                     ;READ COMMAND
1272    000016                  DLRDNH  =16                     ;READ SECTOR NO HEADER CHECK
1273                    ;
1274    000001                  READY   =1                      ;DRIVE READY BIT IN STATUS REG.
1275    000013                  DLSR    =13                     ;STATUS AND RESET
1276    177730                  DLERR   =177730                 ;MASK FOR COVER OPEN
1277    000006                  DLUN    =6                      ;HEADS UNLOADED
1278    000177                  DLCYL   =000177                 ;MASK FOR CYLINDER ADDRESS
1279    100200                  DLDNER  =100200                 ;DONE SET OR ERROR SET BITS
1280
1281    177560                  TTICSR  =       177560          ;KEYBOARD INPUT STATUS
1282    177562                  TTIBFR  =       177562          ;KEYBOARD DATA REGISTER
1283    177564                  TTOCSR  =       177564          ;CONSOLE PRINTER STATUS REGISTER
1284    177566                  TTOBFR  =       177566          ;CONSOLE PRINTER DATA REGISTER
1285
```

```
                                        .SBTTL  SPECIAL MACROS AND OPDEFS.


                                ;+
                                ;SAVE GENERAL REGS 1 TO 5
                                ;-

                                        .MACRO  SAVREG
                                        JSR     R5,REGSAV
                                        .ENDM

                                ;+
                                ; MACRO TO FORCE AN ERROR
                                ;-
                                .MACRO  FORCERROR       TAG,NOTSSR
                                .NLIST
                                .IIF NDF LISTALL, .NLIST
                                .LIST
                                .IF B NOTSSR
                                        MOV     TSSR(R5),R1             ;READ TSSR
                                .ENDC
                                        MOV     FORCER,FORCER           ;IS FORCER SET? (LEAVE C BIT ALONE)
                                        BNE     TAG                     ;BR IF YES
                                .NLIST
                                .IIF NDF LISTALL, .LIST
                                .LIST
                                .ENDM

                                ;+
                                ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
                                ;       WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
                                ;       SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
                                ;       FORCER TO 177777
                                ;       TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
                                ;-
                                .MACRO  FORCEXIT        TAG
                                .NLIST
                                .IIF NDF LISTALL, .NLIST
                                .LIST
                                        MOV     FORCER,FORCER           ;IS FORCER NEGATIVE?
                                        BMI     TAG                     ;BR IF YES
                                .NLIST
                                .IIF NDF LISTALL, .LIST
                                .LIST
                                .ENDM
                                ;+
                                ; MACRO TO INCREMENT ERROR COUNTS
                                ;-
                                .MACRO  NEXT.ERRNO
                                .NLIST
                                ;;;.IIF NDF LISTALL, .NLIST
                                        ERRNO=ERRNO+1
                                ;;;.IIF NDF LISTALL, .LIST
                                .LIST
                                .ENDM

                                ;+
```

```
                                ;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   PR104           ;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                  MESSFA::        .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  <TU80>
003334                          LSDVTYP::
003334  124  125  070                    .ASCIZ  /TU80/
                                        .EVEN


                                ;+
                                ;TEST DESCRIPTION
                                ;
003342                                  DESCRIPT <CZTUWA0 TU80 FRONT END PRT A>
003342                          LSDESC::
003342  103  132  124                    .ASCIZ  /CZTUWA0 TU80 FRONT END PRT A/
                                        .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/
```

```
1565 004021  040  040  125  USI:    .ASCIZ  /  UNEXPECTED INTERRUPT/
1566 004050  040  040  111  NSI:    .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
1567 004113  045  116  045  FNOINTR:        .ASCII  /%N%A/
1568 004117  040  040  116  NOINTR: .ASCIZ  /  NO INTERRUPT WAS GENERATED/
1569 004154  040  040  111  IFAULT: .ASCIZ  /  INTERRUPT FAULT/
1570 004176  045  101  040  INTX:   .ASCIZ  /%A  CPU PC: %06%A  TSBA: %06/
1571 004233  040  040  042  NOINIT: .ASCIZ  /  "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
1572 004305  040  040  042  NSINIT: .ASCIZ  /  "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
1573 004355  040  040  042  BRINIT: .ASCIZ  /  "BUS-RESET" DIDN'T INITIALIZE THE DPU/
1574
1575 004425  000            NUL:    .ASCIZ  //
1576 004426  045  116  000  NULCR:  .ASCIZ  /%N/
1577 004431  045  101  040  EXPGOT: .ASCIZ  /%A EXP'D: %06%A, REC'D: %06/
1578 004465  045  116  045  EXPGT2: .ASCIZ  /%N%A EXP'D: %06%A, %06%N%A REC'D: %0%A, %06/
1579 004541  045  101  040  D'JAD12: .ASCIZ  /%A REG(W) WRITTEN TO: %06%A REG(R) READ; EXP'D: %06%A, REC'D: %06/
1580 004643  122  101  115  PKTRAM::        .ASCIZ  'RAM Contents Do Not Match Packet Sent'
1581 004711  040  040  103  SCME:   .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
1582 004754  127  122  111  WRTMSG: .ASCIZ  'WRITE CHARACTERISTICS Failed'
1583 005011  124  123  123  WRTERR: .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
1584 005104  124  123  123  RDERR:  .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
1585                         .EVEN
1586
1587
1588
```

```
                                        .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.
                                ;--

005176                                  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
```

```
                        .SBTTL  PRITSSR - PRINT TSSR CONTENTS

                ;+
                ;
                ;ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
                ;THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
                ;BY A MESSAGE PRINTING ROUTINE
                ;
                ;INPUTS:
                ;
                ;       R1      CONTENTS OF TSSR
                ;
                ;SUBORDINATE ROUTINES:
                ;
                ;       CHKAMB  CHECK FOR AMBIGUOUS CONTENTS
                ;
                ;-

                PRITSSR:
005264                  SAVREG                          ;SAVE GENERAL REGISTERS
005264 010104           MOV     R1,R4                   ;SAVE THE TSSR CONTENTS
005270                  PRINTB  #TSSRFOR,R4             ;PRINT THE CONTENTS OF TSSR
005272 010446           MOV     R4,-(SP)
005274 012746 005731    MOV     #TSSRFOR,-(SP)
005300 012746 000002    MOV     #2,-(SP)
005304 010600           MOV     SP,R0
005306 104414           TRAP    C$PNTB
005310 062706 000006    ADD     #6,SP
005314 010400           MOV     R4,R0                   ;GET TSSR BACK FOR CHKAMB
005316 004737 016534    JSR     PC,CHKAMB               ;ARE CONTENTS AMBIGUOUS ?
005322 103410           BCS     5$                      ;BRANCH IF NOT
005324                  PRINTX  #AMBTSSR                ;SHOW CONTENTS ARE AMBIGUOUS
005324 012746 006151    MOV     #AMBTSSR,-(SP)
005330 012746 000001    MOV     #1,-(SP)
005334 010600           MOV     SP,R0
005336 104415           TRAP    C$PNTX
005340 062706 000004    ADD     #4,SP
005344 010403           5$:     MOV     R4,R3                   ;CONTENTS OF TSSR
005346 042703 001476            BIC     #HIADDR!FATERR!TERCLS,R3        ;CLEAR ALL MULTIPLE BIT FIELDS
005352 001434           BEQ     20$                     ;NO BITS ARE SET
005354 012702 002576    MOV     #TMPBFR,R2              ;TEMPORARY ASCII BUFFER
005360 012701 003400    MOV     #TSSRBIT,R1             ;ASCII EQUIVALENT OF BITS
005364 005703           10$:    TST     R3              ;REMAINING BITS TO CONVERT
005366 001413           BEQ     15$                     ;BRANCH WHEN ALL ARE DONE
005370 000241           CLC                             ;CLEAR CARRY FOR SHIFT
005372 006103           ROL     R3                      ;SHIFT NEXT BIT TO CARRY
005374 103006           BCC     13$                     ;BRANCH IF BIT NOT SET
005376 011100           MOV     (R1),R0                 ;POINTER TO BIT DEFINITION
005400 112022           11$:    MOVB    (R0)+,(R2)+     ;MOVE ASCIZ TO BUFFER
005402 001376           BNE     11$                     ;MOVE ALL BITS
005404 112762 000054 177777     MOVB    #',,-1(R2)      ;INSERT A COMMA TO TERMINATE
005412 005721           13$:    TST     (R1)+           ;POINT TO NEXT DESCRIPTION
005414 000763           BR      10$                     ;GET THE REMAINING BITS
005416 105042           15$:    CLRB    -(R2)           ;TERMINATE THE LINE
005420                  PRINTX  #TSSDEF,#TMPBFR         ;PRINT THE BIT DEFINITIONS
005420 012746 002576    MOV     #TMPBFR,-(SP)
005424 012746 006122    MOV     #TSSDEF,-(SP)
```

```
        005430  012746  000002                  MOV     #2,-(SP)
        005434  010600                           MOV     SP,R0
        005436  104415                           TRAP    C$PNTX
        005440  062706  000006                  ADD     #6,SP
   1660
   1661 005444  010403                  20$:    MOV     R4,R3                   ;GET THE TSSR CONTENTS
   1662 005446  042703  177761                  BIC     #^CTERCLS,R3            ;CLEAR ALL BUT TERMINATION
   1663 005452  016303  006212                  MOV     TCOCOD(R3),R3           ;GET THE TERMINATION CODE MEANING
   1664 005456                                   PRINTX  #TCOASC,R3              ;PRINT THE TERMINATION CODE
        005456  010346                           MOV     R3,-(SP)
        005460  012746  006012                  MOV     #TCOASC,-(SP)
        005464  012746  000002                  MOV     #2,-(SP)
        005470  010600                           MOV     SP,R0
        005472  104415                           TRAP    C$PNTX
        005474  062706  000006                  ADD     #6,SP
   1665 005500  010403                          MOV     R4,R3                   ;TSSR CONTENTS AGAIN
   1666 005502  042703  177717                  BIC     #^CFATERR,R3            ;CLEAR ALL BUT FATAL TERMINATION
   1667 005506  001421                          BEQ     25$                     ;DON'T PRINT IF ZERO
   1668 005510  006203                          ASR     R3
   1669 005512  006203                          ASR     R3
   1670 005514  006203                          ASR     R3                      ;ALINE TERMINATION CODE FOR INDEX
   1671 005516  016303  006552                  MOV     TSFCOD(R3),R3           ;GET THE FATAL TERMINATION CODE
   1672 005522                                   PRINTX  #TFCASC,R3              ;PRINT THE FATAL TERMINATION CODE
        005522  010346                           MOV     R3,-(SP)
        005524  012746  006053                  MOV     #TFCASC,-(SP)
        005530  012746  000002                  MOV     #2,-(SP)
        005534  010600                           MOV     SP,R0
        005536  104415                           TRAP    C$PNTX
        005540  062706  000006                  ADD     #6,SP
   1673 005544  012737  000031  002170          MOV     #25.,FATFLG             ;DROP UNIT AFTER THIS ERROR
   1674 005552  010403                  25$:    MOV     R4,R3                   ;GET TSSR CONTENTS
   1675 005554  042703  176277                  BIC     #^CHIADDR,R3            ;CLEAR ALL BUT EXTENDED ADDRESS
   1676 005560  001411                          BEQ     30$                     ;DON'T PRINT IF ZERO
   1677 005562                                   PRINTX  #TEXASC,R3              ;PRINT THE EXTENDED ADDRESS BITS
        005562  010346                           MOV     R3,-(SP)
        005564  012746  005751                  MOV     #TEXASC,-(SP)
        005570  012746  000002                  MOV     #2,-(SP)
        005574  010600                           MOV     SP,R0
        005576  104415                           TRAP    C$PNTX
        005600  062706  000006                  ADD     #6,SP
   1678 005604  022704  100210          30$:    CMP     #100210,R4              ;CHECK FOR MEDIA ERROR
   1679 005610  001003                          BNE     31$                     ;BR, IF PROBABLY NOT TAPE ERROR
   1680 005612  012737  005672  002146          MOV     #EPRT3,EPRTSW           ;"PROBABLY MEDIA RELETED ERROR - BAD TAPE"
   1681 005620  005737  002146          31$:    TST     EPRTSW                  ;CHECK FOR THE SWITCH EMPTY
   1682 005624  001003                          BNE     310$                    ;BR, IF SWITCH IS NOT EMPTY
   1683 005626  012737  005672  002146          MOV     #EPRT1,EPRTSW           ;SET SWITCH TO DEFAULT
   1684 005634  013737  002146  005644  310$:   MOV     EPRTSW,32$+2            ;PUT REAL SWITCHABLE MESSAGE IN PLACE
   1685 005642                          32$:    PRINTB  #EPRT1                  ;PRINT THE ERPOR MESSAGE
        005642  012746  005672                  MOV     #EPRT1,-(SP)
        005646  012746  000001                  MOV     #1,-(SP)
        005652  010600                           MOV     SP,R0
        005654  104414                           TRAP    C$PNTB
        005656  062706  000004                  ADD     #4,SP
   1686 005662  012737  005672  002146          MOV     #EPRT1,EPRTSW           ;RESET TO NORMAL ERROR POINTER
   1687 005670  000207                          RTS     PC                      ;RETURN TO CALLER
   1688
   1689 005672                          EPRT2:
```

```
1690 005672                          EPRT3:
1691 005672     045     116     045  EPRT1:  .ASCIZ  '%N%A *****REPLACE M7454*****%S'
1692 005731     045     116     045  TSSRFOR:        .ASCIZ  '%N%A TSSR = %O6'
1693 005751     045     116     045  TEXASC:         .ASCIZ  '%N%A Extended Address Bits = %O6'
1694 006012     045     116     045  TCOASC:         .ASCIZ  '%N%A Termination Class Code = %T'
1695 006053     045     116     045  TFCASC:         .ASCIZ  '%N%A Fatal Termination Class Code = %T'
1696 006122     045     116     045  TSSDEF:         .ASCIZ  '%N%A TSSR Bits Set: %T'
1697 006151     045     116     045  AMBTSSR:        .ASCIZ  '%N%A TSSR Contents Are Ambiguous'
1698                                  .EVEN
1699 006212  006232  006255  006303  TCOCOD: .WORD   1$,2$,3$,4$,5$,6$,7$,8$
1700 006232     116     157     162  1$:     .ASCIZ  'Normal Termination'
1701 006255     124     145     162  2$:     .ASCIZ  'Termination Condition'
1702 006303     124     141     160  3$:     .ASCIZ  'Tape Status Alert'
1703 006325     106     165     156  4$:     .ASCIZ  'Function Reject'
1704 006345     122     145     143  5$:     .ASCIZ  'Recoverable Error - Tape Position One Record Down'
1705 006427     122     145     143  6$:     .ASCIZ  'Recoverable Error - Tape Was Not Moved'
1706 006476     125     156     162  7$:     .ASCIZ  'Unrecoverable Error'
1707 006522     106     141     164  8$:     .ASCIZ  'Fatal Controller Error'
1708                                  .EVEN
1709
1710 006552  006562  006616  006627  TSFCOD: .WORD   1$,2$,3$,4$
1711 006562     111     156     164  1$:     .ASCIZ  'Internal Diagnostic Failure'
1712 006616     122     145     163  2$:     .ASCIZ  'Reserved'
1713 006627     102     165     163  3$:     .ASCIZ  'Bus Interface or Sanity Check Error'
1714 006673     122     145     163  4$:     .ASCIZ  'Reserved'
1715                                  .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.
                                ;-
006704                          PRIPKT::
006704                                  SAVREG                          ;SAVE THE REGISTERS
006710  010005                          MOV     R0,R5                   ;SAVE NO. OF WORDS IN PACKET
006712  005737  003102                  TST     KTENABLE                ;ABOVE 28K UNDER TEST?
006716  001001                          BNE     10$                     ;BR IF YES
006720  005003                          CLR     R3                      ;SET HIGH ORDER ADDRESS TO 0
006722  010301                  10$:    MOV     R3,R1                   ;COPY HIGH ORDER ADDRESS
006724  010400                          MOV     R4,R0                   ;GET LOWER ADDRESS
006726  006100                          ROL     R0                      ;SHIFT BIT 15 INTO C BIT
006730  006101                          ROL     R1                      ;AND INTO HIGH ORDER.
006732                                  PRINTB  #PKTADD,R1,R4           ;PRINT PACKET ADDRESS
006732  010446                          MOV     R4,-(SP)
006734  010146                          MOV     R1,-(SP)
006736  012746  007110                  MOV     #PKTADD,-(SP)
006742  012746  000003                  MOV     #3,-(SP)
006746  010600                          MOV     SP,R0
006750  104414                          TRAP    C$PNTB
006752  062706  000010                  ADD     #10,SP
006756  010300                  15$:    MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
006760  001404                          BEQ     20$                     ;BR IF NOT ABOVE 28K.
006762  010401                          MOV     R4,R1                   ;GET LOW ORDER ADDRESS
006764  004737  020106                  JSR     PC,SETMAP               ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
006770  010004                          MOV     R0,R4                   ;GET RETURNED PAR6 ADDRESS BIAS
006772  005001                  20$:    CLR     R1                      ;SAVE WORD NUMBER
006774  012402                  25$:    MOV     (R4)+,R2                ;GET PACKET CONTENTS
006776                                  PRINTB  #PKTFRM,R1,R2           ;PRINT THE DATA
006776  010246                          MOV     R2,-(SP)
007000  010146                          MOV     R1,-(SP)
007002  012746  007052                  MOV     #PKTFRM,-(SP)
007006  012746  000003                  MOV     #3,-(SP)
007012  010600                          MOV     SP,R0
007014  104414                          TRAP    C$PNTB
007016  062706  000010                  ADD     #10,SP
007022  005201                          INC     R1                      ;NEXT WORD NUMBER
007024  020105                          CMP     R1,R5                   ;DONE ALL PACKET WORDS?
007026  002762                          BLT     25$                     ;LOOP TILL ALL DONE
007030                                  PRINTB  #PKTNEW                 ;JUST A COUPLE NEW LINES
007030  012746  007145                  MOV     #PKTNEW,-(SP)
007034  012746  000001                  MOV     #1,-(SP)
007040  010600                          MOV     SP,R0
007042  104414                          TRAP    C$PNTB
007044  062706  000004                  ADD     #4,SP
007050  000207                          RTS     PC                      ;RETURN
007052     045     116     045  PKTFRM: .ASCIZ  '%N%A Packet Word #%D1%A = %O6'
007110     045     116     045  PKTADD: .ASCIZ  '%N%A Packet Address = %O1%O5'
007145     045     116     045  PKTNEW: .ASCIZ  '%N%N%A '
                                        .EVEN
```

```
                                        .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
                                ;
                                ;-

007156                          PRIBXOR::
007156                                  SAVREG                          ;SAVE THE REGISTERS
007162  010203                          MOV     R2,R3                   ;EXPECTED DATA
007164                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007174  012700  177400                  MOV     #^C<377>,R0             ;BYTE MASK
007200  040001                          BIC     R0,R1                   ;SAVE LOW BYTE RECV
007202  040002                          BIC     R0,R2                   ;SAVE LOW BYTE EXPD
007204  040003                          BIC     R0,R3                   ;SAVE LOW BYTE XOR
007206                                  PRINTB  #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
007206  010346                          MOV     R3,-(SP)
007210  010146                          MOV     R1,-(SP)
007212  010246                          MOV     R2,-(SP)
007214  012746  007240                  MOV     #XORBFOR,-(SP)
007220  012746  000004                  MOV     #4,-(SP)
007224  010600                          MOV     SP,R0
007226  104414                          TRAP    C$PNTB
007230  062706  000012                  ADD     #12,SP
007234  010300                          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007236  000207                          RTS     PC                      ;RETURN TO CALLER

007240      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
                        ;
                        ;-

007306                  PRIXOR::
007306                          SAVREG                          ;SAVE THE REGISTERS
007312  010203                  MOV     R2,R3                   ;EXPECTED DATA
007314                          XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007324                          PRINTB  #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
007324  010346                  MOV     R3,-(SP)
007326  010146                  MOV     R1,-(SP)
007330  010246                  MOV     R2,-(SP)
007332  012746  007356          MOV     #XORFOR,-(SP)
007336  012746  000004          MOV     #4,-(SP)
007342  010600                  MOV     SP,R0
007344  104414                  TRAP    C$PNTB
007346  062706  000012          ADD     #12,SP
007352  010300                  MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007354  000207                  RTS     PC                      ;RETURN TO CALLER

007356     045     116     045  XORFOR: .ASCIZ  '%N%A EXPD: %O6%A RECV: %O6%A XOR: %O6'
                                .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
                                ;
                                ;-

007424                          PRIEQU:
007424                                  SAVREG                          ;SAVE THE REGISTERS
007430  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
                                ;
                                ;-
007432                          PRIRAM:
007432                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
007436                                  PRINTB  #RAMFOR,R4              ;PRINT RAM ADDRESS IN ERROR
007436  010446                          MOV     R4,-(SP)
007440  012746  007462                  MOV     #RAMFOR,-(SP)
007444  012746  000002                  MOV     #2,-(SP)
007450  010600                          MOV     SP,R0
007452  104414                          TRAP    C$PNTB
007454  062706  000006                  ADD     #6,SP
007460  000207                          RTS     PC                      ;RETURN

007462    045   116   045   RAMFOR: .ASCIZ  '%N%A CONTROLLER RAM ADDRESS = %06'
                                        .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

1871                                    ;
1872                                    ;-
1873 007524                             PRIADD:
1874 007524                                     SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
1875 007530  013700  002202                     MOV     ERRHI,R0                ;GET HIGH ADDRESSS
1876 007534  013701  002204                     MOV     ERRLO,R1                ;GET LOW ADDRESS
1877 007540  010102                             MOV     R1,R2                   ;COPY LOW ADDRESS
1878 007542  006101                             ROL     R1                      ;SHIFT BIT 15 TO C BIT
1879 007544  006100                             ROL     R0                      ;SHIFT INTO HIGH ORDER
1880 007546                                     PRINTB  #PRIA0,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
     007546  010246                             MOV     R2,-(SP)
     007550  010046                             MOV     R0,-(SP)
     007552  012746  007574                     MOV     #PRIA0,-(SP)
     007556  012746  000003                     MOV     #3,-(SP)
     007562  010600                             MOV     SP,R0
     007564  104414                             TRAP    C$PNTB
     007566  062706  000010                     ADD     #10,SP
1881 007572  000207                             RTS     PC                      ;RETURN
1882
1883 007574    045    116    045  PRIA0:      .ASCIZ  '%N%A MEMORY ERROR ADDRESS = %01%05'
1884                                            .EVEN
1885
1886
1887                                            .SBTTL  PRITADD - PRINT MEMORY TEST ADDRESS
1888                                    ;+
1889                                    ;
1890                                    ;PRINT MEMORY ADDRESS
1891                                    ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
1892                                    ;
1893                                    ; IMPLICIT INPUTS
1894                                    ;
1895                                    ;       ERRHI   - HIGH ORDER ADDRESS
1896                                    ;       ERRLO   - LOW ORDER ADDRESS
1897                                    ;
1898                                    ;-
1899 007640                             PRITADD:
1900 007640                                     SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
1901 007644  013700  002202                     MOV     ERRHI,R0                ;GET HIGH ADDRESSS
1902 007650  013701  002204                     MOV     ERRLO,R1                ;GET LOW ADDRESS
1903 007654  010102                             MOV     R1,R2                   ;COPY LOW ADDRESS
1904 007656  006101                             ROL     R1                      ;SHIFT BIT 15 TO C BIT
1905 007660  006100                             ROL     R0                      ;SHIFT INTO HIGH ORDER
1906 007662                                     PRINTB  #PRIT0,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
     007662  010246                             MOV     R2,-(SP)
     007664  010046                             MOV     R0,-(SP)
     007666  012746  007710                     MOV     #PRIT0,-(SP)
     007672  012746  000003                     MOV     #3,-(SP)
     007676  010600                             MOV     SP,R0
     007700  104414                             TRAP    C$PNTB
     007702  062706  000010                     ADD     #10,SP
1907 007706  000207                             RTS     PC                      ;RETURN
1908 007710    045    116    045  PRIT0:      .ASCIZ  '%N%A MEMORY TEST ADDRESS = %01%05'
1909                                            .EVEN
```

```
                                        .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:
                                ;
                                ;
                                ;-

007752                          SPACE::
007752                                  SAVREG                  ;SAVE THE GENERAL REGISTERS
007756  012737  000764  010150          MOV     #500.,SDELAY    ;SET UP DELAY
007764  012737  140010  010140          MOV     #140010,80$     ;SET UP COMMAND, SPACE FORWARD
007772  005703                          TST     R3              ;CHECK FOR DIRECTION
007774  100403                          BMI     5$              ;BR, IF REVERSE INDICATED
007776  010337  010142                  MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010002  000407                          BR      10$             ;GO DO COMMAND
010004  042703  100000          5$:     BIC     #BIT15,R3       ;CLEAR DIRECTION BIT
010010  010337  010142                  MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010014  052737  000400  010140          BIS     #BIT8,80$       ;SET REVERSE BIT IN COMMAND PACKET
010022  012704  010140          10$:    MOV     #80$,R4         ;SET UP R4 WITH PACKET ADDRESS
010026  010465  177776                  MOV     R4,TSDB(R5)     ;SEND OUT COMMAND
010032  004737  016740          15$:    JSR     PC,WAITF        ;WAIT FOR SSR
010036  103420                          BCS     20$             ;BR, IF SSR IS SET AND OK
010040                                  DELAY   250             ;DELAY ABOUT .25 SECONDS
010040  012727  000250                  MOV     #250,(PC)+
010044  000000                          .WORD   0
010046  013727  002116                  MOV     L$DLY,(PC)+
010052  000000                          .WORD   0
010054  005367  177772                  DEC     -6(PC)
010060  001375                          BNE     .-4
```

```
        010062  005367  177756                  DEC     -22(PC)
        010066  001367                          BNE     .-20
   1962 010070  005337  010150                  DEC     SDELAY          ;BUMP DELAY COUNTER DOWN
   1963 010074  001356                          BNE     15$             ;BR, IF MORE DELAY
   1964 010076  000411                          BR      60$             ;BR IF TROUBLE CARRY = CLEAR
   1965 010100  016501  000000          20$:    MOV     TSSR(R5),R1     ;READ TSSR
   1966 010104  012702  000200                  MOV     #SSR,R2         ;SET UP EXPECTED
   1967 010110  020201                  25$:    CMP     R2,R1           ;ARE THEY OK
   1968 010112  001401                          BEQ     40$             ;BR, IF EQUAL = OK
   1969 010114  000402                          BR      60$             ;TROUBLE EXIT
   1970 010116  000261                  40$:    SEC                     ;SET CARRY NO TROUBLE
   1971 010120  000401                          BR      70$             ;EXIT
   1972 010122  000241                  60$:    CLC                     ;CARRY CLEAR = ERROR
   1973 010124                          70$:
   1974 010124  010400                          MOV     R4,R0           ;PASS PACKET ADDRESS
   1975 010126  000207                          RTS     PC              ;RETURN
   1976                                 ;
   1977                                 ;PACKET FOR SPACE COMMAND
   1978                                 ;
   1980 010130                                  .BLKB   10-<.-TUV2A&7>
   1982                                 ;
   1983                                 ;COMMAND WORD
   1984 010140  000000                  80$:    .WORD
   1985                                 ;NUMBER OF RECORDS TO BE SPACED OVER WORD
   1986 010142  000000                  90$:    .WORD
   1987 010144  000000                          .WORD
   1988 010146  000000                          .WORD
   1989 010150  000000                  SDELAY: .WORD   0               ;DELAY COUNTER
   1990                                         .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:
                        ;
                        ;
                        ;-

                        WRTCHR::
010152                          SAVREG                  ;SAVE THE GENERAL REGISTERS
010152  005037  002174          CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
010156  010465  177776  10$:    MOV     R4,TSDB(R5)     ;SEND OUT COMMAND
010162  004737  017054          JSR     PC,CHKTSSR      ;WAIT FOR SSR
010166  103401                  BCS     20$             ;BR, IF SSR IS SET AND OK
010170  000423                  BR      60$             ;BR IF TROUBLE CARRY = CLEAR
010172  016501  000000  20$:    MOV     TSSR(R5),R1     ;READ TSSR
010176  012702  000200          MOV     #SSR,R2         ;SET UP EXPECTED
010202  032701  000100          BIT     #OFL,R1         ;WAS OFF LINE SET IN TSSR
010206  001402                  BEQ     25$             ;BR, IF NO OFL SET
010210  052702  000100          BIS     #OFL,R2         ;MAKE THEM LOOK ALIKE
010214  020201  25$:    CMP     R2,R1           ;ARE THEY OK
010216  001401                  BEQ     40$             ;BR, IF EQUAL = OK
010220  000407                  BR      60$             ;TROUBLE EXIT
010222  062704  000010  40$:    ADD     #8.,R4          ;POINT TO WRT CHARA DATA PACKET
010226  011403                  MOV     (R4),R3         ;GET ADDRESS OF MESSAGE BUFFER
010230  010337  002716          MOV     R3,MESBFA       ;STORE FOR PRINT ROUTINES
010234  000261                  SEC                     ;SET CARRY NO TROUBLE
010236  000401                  BR      70$             ;EXIT
010240  000241  60$:    CLC                     ;CARRY CLEAR = ERROR
010242  016500  000000  70$:    MOV     TSSR(R5),R0     ;RETURN TSSR CONTENTS
010246  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::
010254                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
010254 010260  012704  010350           MOV     #RWPACK,R4      ;GET PACKET ADDRESS
010264  010465  177776                  MOV     R4,TSDB(R5)     ;SEND PACKET ADDRESS TO EXECUTE
010270  012703  000550                  MOV     #360.,R3        ;ENOUGH TIME FOR 2400' REEL TO REWIND
010274  004737  016740          10$:    JSR     PC,WAITF        ;WAIT FOR SSR TO SET
010300  103417                          BCS     20$             ;LEAVE WHEN SSR IS SET
010302                                  DELAY   250.            ;WAIT FOR .25 SECONDS
010302  012727  000372                  MOV     #250.,(PC)+
010306  000000                          .WORD   0
010310  013727  002116                  MOV     L$DLY,(PC)+
010314  000000                          .WORD   0
010316  005367  177772                  DEC     -6(PC)
010322  001375                          BNE     .-4
010324  005367  177756                  DEC     -22(PC)
010330  001367                          BNE     .-20
010332  005303                          DEC     R3              ;BUMP COUNTER DOWN
010334  001357                          BNE     10$             ;KEEP GOING
010336  000241                          CLC                     ;CLEAR CARRY TO SET ERROR
010340  010400                  20$:    MOV     R4,R0           ;PASS THE PACKET ADDRESS
010342  000207                          RTS     PC              ;RETURN


010344                                  .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  017054          JSR     PC,CHKTSSR      ;WAIT FOR SSR
010376  004737  017054  10$:    JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
010402  110265  177777          MOVB    R2,TSDBH(R5)    ;SELECT NEXT RAM ADDRESS
010406  004737  017054          JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
010412  116511  177776          MOVB    TSBAL(R5),(R1)  ;READ THE RAM DATA
010416  122124                  CMPB    (R1)+,(R4)+     ;COMPARE TO EXPECTED
010420  001401                  BEQ     20$             ;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  017054          10$:    JSR     PC,CHKTSSR              ;WAIT FOR THE SSR TO SET
010506  110265  177777                  MOVB    R2,TSDBH(R5)            ;SELECT NEXT RAM ADDRESS
010512  004737  017054                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
010516  116521  177776                  MOVB    TSBAL(R5),(R1)+         ;READ THE RAM DATA
010522  062702  000001          20$:    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                                  PPINTX  #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
010636  000207                  50$:    RTS     PC                      ;RETURN
010640  000000                  RAMHLD: .WORD   0                       ;RAM ADDR HOLDER 1ST ADDRESS
010642  000000                  RAMR5H: .WORD   0                       ;HOLDS R5 FOR LATER
010644     045     116     045  RAMIOP: .ASCIZ  '%N%A Ram Address (Octal) = %03%A - %03%N'
010715     045     101     040  RAMPD:  .ASCIZ  '%A %03%A '
                                        .EVEN
```

```
                                        .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  017054                  JSR     PC,CHKTSSR      ;WAIT FOR SSR
010752  004737  017054          10$:    JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
010756  110265  177777                  MOVB    R2,TSDBH(R5)    ;SELECT NEXT RAM ADDRESS
010762  004737  017054                  JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
010766  116511  177776                  MOVB    TSBAL(R5),(R1)  ;READ THE RAM DATA
010772  122124                          CMPB    (R1)+,(R4)+     ;COMPARE TO EXPECTED
010774  001401                          BEQ     20$             ;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  020106                  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
2310
```

```
                                        .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
                                ;
                                ;-
                                CKMSG2::
                                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011156  020327  000144                  CMP     R3,#RECMSG-EXPMSG;aad IS COUNT ABOVE MAX ALLOWED?
011162  003412                         BLE     5$                      ;aad BR IF NO
011164  012703  000144                  MOV     #RECMSG-EXPMSG,R3;aad
011170                                  PRINTF  #DEBUGMSG               ;aad
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  020106                  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
```

```
2364 011274  000241                          CLC                     ;SET FAILURE
2365 011276  000401                          BR      60$             ;
2366 011300  000261                  55$:    SEC                     ;SET SUCCESS
2367 011302  000207                  60$:    RTS     PC              ;RETURN
2368
2369 011304      120     122     117  DEBUGMSG:       .ASCIZ  'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-'<200>
2370 011374      045     116     045  FERCM:  .ASCII  /%N%A  ***/
2371 011405      040     040     124  ERCM:   .ASCIZ  /  TSSR ERROR CODE REC'D = /
2372 011440      056     056     056  SIMSG:  .ASCIZ  /.... AFTER DOING SOFT INIT/
2373 011473      124     105     123  TINERR: .ASCIZ  /TEST: .../
2374                                  .EVEN
```

```
2376
2377
2378                                    ;+
2379                                    ;
2380                                    ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
2381                                    ;
2382                                    ;INPUT:
2383                                    ;
2384                                    ;       R1      CONTENTS OF TSSR AT ERROR
2385                                    ;
2386                                    ;SIDE EFFECTS:
2387                                    ;
2388                                    ;       EXECUTES DROP UNIT TO CEASE TESTING
2389                                    ;
2390                                    ;-
2391
2392 011506                                    BGNMSG  SFIMSG
     011506                             SFIMSG::
2393 011506  004737  005264                    JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
2394 011512  004737  017772                    JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
2395 011516                                    ENDMSG
     011516                             L10003:
     011516  104423                            TRAP    C$MSG
2396
2397                                    ;+
2398                                    ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2399                                    ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
2400                                    ;
2401                                    ;INPUTS:
2402                                    ;
2403                                    ;       R1      TSSR CONTENTS
2404                                    ;       R4      ADDRESS OF COMMAND PACKET
2405                                    ;
2406                                    ;-
2407
2408 011520                                    BGNMSG  PKTSSR
     011520                             PKTSSR::
2409 011520  004737  005264                    JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
2410 011524  012700  000004                    MOV     #4,R0           ;NO. OF WORDS IN PACKET
2411 011530  004737  006704                    JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
2412 011534  013700  002716                    MOV     MESBFA,R0       ;ADDRESS OF MESSAGE BUFFER
2413 011540  005001                            CLR     R1              ;ASSUME NO HIGH MEMORY
2414 011542  004737  013702                    JSR     PC,PRMESS       ;PRINT THE MESSAGE BUFFER ALSO
2415 011546                                    ENDMSG
     011546                             L10004:
     011546  104423                            TRAP    C$MSG
2416
2417                                    ;+
2418                                    ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2419                                    ;TSSR AND A GET STATUS COMMAND PACKET.
2420                                    ;
2421                                    ;INPUTS:
2422                                    ;
2423                                    ;       R1      TSSR CONTENTS
2424                                    ;       R4      ADDRESS OF COMMAND PACKET
2425                                    ;
2426                                    ;-
```

```
2427
2428 011550                                    BGNMSG  PKTGETS
     011550                                PKTGETS::
2429 011550  004737  005264                    JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
2430 011554  012700  000002                    MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
2431 011560  004737  006704                    JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
2432 011564                                    ENDMSG
     011564                                L10005:
     011564  104423                            TRAP    C$MSG
2433
2434
2435                                        ;+
2436                                        ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
2437                                        ;
2438                                        ;INPUTS:
2439                                        ;
2440                                        ;       R1      TSSR CONTENTS
2441                                        ;       R4      ADDRESS OF COMMAND PACKET
2442                                        ;-
2443
2444 011566                                    BGNMSG  SFFMSG
     011566                                SFFMSG::
2445 011566  004737  005264                    JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
2446 011572                                    ENDMSG
     011572                                L10006:
     011572  104423                            TRAP    C$MSG
2447
2448
2449                                            .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
2450                                        ;+
2451                                        ;
2452                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
2453                                        ;BUFFER FOR ERROR REPORTS
2454                                        ;
2455                                        ;INPUTS:
2456                                        ;
2457                                        ;       R1      CONTENTS OF TSSR
2458                                        ;       R2      LOW ORDER MESSAGE BUFFER
2459                                        ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
2460                                        ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
2461                                        ;-
2462 011574                                    BGNMSG  PKTMES
     011574                                PKTMES::
2463 011574  004737  005264                    JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR
2464 011600  010200                            MOV     R2,R0           ;LOW ORDER ADDRESS
2465 011602  010301                            MOV     R3,R1           ;HIGH ORDER ADDRESS
2466 011604  004737  013702                    JSR     PC,PRMESS       ;PRINT THE MESSAGE BUFFER
2467 011610                                    ENDMSG
     011610                                L10007:
     011610  104423                            TRAP    C$MSG
2468
```

```
                                        .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
011612                          ADDSSR::
011612
011612  004737  007640                  JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
011616  016501  000000                  MOV     TSSR(R5),R1     ;GET CURRENT TSSR
011622  004737  005264                  JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
011626                                  ENDMSG
011626                          L10010:
011626  104423                          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
011630                          MSGEXP::
011630
011630  012700  000007                  MOV     #7,R0           ;ASSUME NO EXT FEATURES
011634  004737  015242          5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
011640                                  ENDMSG
011640                          L10011:
011640  104423                          TRAP    C$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
                                        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  015612                          JSR     PC,PRBYTEXP     ;PRINT FIFO BYTES IN ERROR
                                                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
                                ;-
                                        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$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
012056  004737  015242                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012062                                  ENDMSG
012062                          L10013:
012062  104423                          TRAP    C$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
                                ;-
                                        BGNMSG  MSGLOOP
012536                          MSGLOOP::
012536  012701  012600                  MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
```

```
CZTUWA0 TU80 FRONT END PRT A    MACRO M1200  29-MAR-83 13:24  PAGE 38-1
MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

   2579 012542  012100                 10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
   2580 012544  001410                         BEQ     20$             ;BR IF YES
   2581 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$PNTX
        012560  062706  000004                 ADD     #4,SP
   2582 012564  000766                         BR      10$             ;DO ANOTHER MSG LINE
   2583 012566  012700  000012           20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
   2584 012572  004737  015242                 JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
   2585 012576                                 ENDMSG
        012576                          L10014:
        012576  104423                         TRAP    C$MSG
   2586
   2587 012600  012620  012673  012772  LOOPCOD:        .WORD   1$,2$,3$,4$,5$,6$,7$,0
   2588 012620     045     116     045  1$:.ASCIZ  '%N%A Tape Bus Loopback Signals in Word #8:'
   2589 012673     045     116     045  2$:.ASCIZ  '%N%A      PARERR<15>          IRESV2<14>          IRESV1<13>'
   2590 012772     045     116     045  3$:.ASCIZ  '%N%A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
   2591 013071     045     116     045  4$:.ASCIZ  '%N%A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
   2592 013170     045     116     045  5$:.ASCIZ  '%N%A ITAD0=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
   2593 013267     045     116     045  6$:.ASCIZ  '%N%A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
   2594 013366     045     116     045  7$:.ASCIZ  '%N%A IGO  =>IFPT<00>'
   2595                                         .EVEN
   2596
```

```
                                                        .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
013414  012700  000012                          MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
013420  004737  015242                          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
013426  004737  007524                          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  007306                          JSR     PC,PRIXOR       ;PRINT EXPD/RECV
013446                                          ENDMSG
013446                                          L10016:
013446  104423                                  TRAP    C$MSG
```

```
                                        .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
                                ;TU80.
                                ;
                                ;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$:    MOV     R1,R3                   ;SAVE HIGH ORDER ADDRESS
013726  006100                          ROL     R0                      ;SHIFT BIT15 TO C BIT
013730  006101                          ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
013732                                  PRINTX  #PROASC,R1,R5           ;PRINT MESSAGE BUFFER ADDRESS
013732  010546                          MOV     R5,-(SP)
013734  010146                          MOV     R1,-(SP)
013736  012746  014533                  MOV     #PROASC,-(SP)
013742  012746  000003                  MOV     #3,-(SP)
013746  010600                          MOV     SP,R0
013750  104415                          TRAP    C$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  014453                  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  #PR1AS,                 ;PRINT HEADER FOR CONTENTS
014004  012746  014600                  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  020106                  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  014636                  MOV     #MESHEA,-(SP)
        014050  012746  000002                  MOV     #2,-(SP)
        014054  010600                          MOV     SP,R0
        014056  104415                          TRAP    C$PNTX
        014060  062706  000006                  ADD     #6,SP
  2728  014064                                  PRINTX  #DATAFL,(R5)+   ;PRINT "DATA FIELD LENGTH     ="
        014064  012546                          MOV     (R5)+,-(SP)
        014066  012746  014703                  MOV     #DATAFL,-(SP)
        014072  012746  000002                  MOV     #2,-(SP)
        014076  010600                          MOV     SP,R0
        014100  104415                          TRAP    C$PNTX
        014102  062706  000006                  ADD     #6,SP
  2729  014106                                  PRINTX  #RBPCRA,(R5)+   ;PRINT "RESIDUAL BYTE COUNTER ="
        014106  012546                          MOV     (R5)+,-(SP)
        014110  012746  014750                  MOV     #RBPCRA,-(SP)
        014114  012746  000002                  MOV     #2,-(SP)
        014120  010600                          MOV     SP,R0
        014122  104415                          TRAP    C$PNTX
        014124  062706  000006                  ADD     #6,SP
  2730  014130                                  PRINTX  #XSOCON,(R5)+   ;PRINT "XSTAT0 CONTENTS       ="
        014130  012546                          MOV     (R5)+,-(SP)
        014132  012746  015015                  MOV     #XSOCON,-(SP)
        014136  012746  000002                  MOV     #2,-(SP)
        014142  010600                          MOV     SP,R0
        014144  104415                          TRAP    C$PNTX
        014146  062706  000006                  ADD     #6,SP
  2731  014152                                  PRINTX  #XS1CON,(R5)+   ;PRINT "XSTAT1 CONTENTS       ="
        014152  012546                          MOV     (R5)+,-(SP)
        014154  012746  015062                  MOV     #XS1CON,-(SP)
        014160  012746  000002                  MOV     #2,-(SP)
        014164  010600                          MOV     SP,R0
        014166  104415                          TRAP    C$PNTX
        014170  062706  000006                  ADD     #6,SP
  2732  014174                                  PRINTX  #XS2CON,(R5)+   ;PRINT "XSTAT2 CONTENTS       ="
        014174  012546                          MOV     (R5)+,-(SP)
        014176  012746  015127                  MOV     #XS2CON,-(SP)
        014202  012746  000002                  MOV     #2,-(SP)
        014206  010600                          MOV     SP,R0
        014210  104415                          TRAP    C$PNTX
        014212  062706  000006                  ADD     #6,SP
  2733  014216                                  PRINTX  #XS3CON,(R5)+   ;PRINT "XSTAT3 CONTENTS       ="
        014216  012546                          MOV     (R5)+,-(SP)
        014220  012746  015174                  MOV     #XS3CON,-(SP)
        014224  012746  000002                  MOV     #2,-(SP)
        014230  010600                          MOV     SP,R0
        014232  104415                          TRAP    C$PNTX
        014234  062706  000006                  ADD     #6,SP
  2734  014240  022737  000001  002134          CMP     #1,TRANSTST     ;CHECK SOFTWARE P-TABLE
  2735  014246  001402                          BEQ     40$             ;DO DUMP
  2736  014250  000137  014360                  JMP     50$             ;DON'T DO THE DUMP
  2737  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
```

```
2738 014274  012737  000010  002246          MOV     #8.,RAMSIZ      ;RAM FIELD IS 8 BYTES LONG
2739 014302  012737  000020  010640          MOV     #20,RAMHLD      ;FIELD STARTS AT 20 OCTAL (10 HEX)
2740 014310  004737  010456                  JSR     PC,RAMER        ;READ AND PRINT THEM
2741 014314  012737  000040  010640          MOV     #40,RAMHLD      ;FIELD STARTS AT 40 OCTAL (20 HEX)
2742 014322  004737  010456                  JSR     PC,RAMER        ;READ AND PRINT THEM
2743 014326  012737  000060  010640          MOV     #60,RAMHLD      ;FIELD STARTS AT 60 OCTAL (30 HEX)
2744 014334  004737  010456                  JSR     PC,RAMER        ;READ AND PRINT THEM
2745 014340  012737  000020  002246          MOV     #16.,RAMSIZ     ;RAM FIELD IS SIXTEEN BYTES LONG
2746 014346  012737  000100  010640          MOV     #100,RAMHLD     ;FIELD STARTS AT 100 OCTAL (40 HEX)
2747 014354  004737  010456                  JSR     PC,RAMER        ;READ AND PRINT THEM
2748 014360  000207                  50$:    RTS     PC              ;RETURN
2749 014362     045     116     045  RAMFHR: .ASCIZ   '%N%A ********** SPECIAL M7434 RAM MEMORY DUMP **********'
2750 014453     045     116     045  MESBFN: .ASCIZ   '%N%A MESSAGE BUFFER CONTENTS PROBABLY NOT VALID'
2751 014533     045     116     045  PROASC: .ASCIZ   '%N%A Message Buffer Address = %01%05'
2752 014600     045     116     045  PR1ASC: .ASCIZ   '%N%A Message Buffer Contents:'
2753
2754 014636     045     116     045  MESHEA: .ASCIZ   '%N%A Message Buffer Header     = %06'
2755 014703     045     116     045  DATAFL: .ASCIZ   '%N%A Data Field Length         = %06'
2756 014750     045     116     045  RBPCRA: .ASCIZ   '%N%A Residual Byte Counter     = %06'
2757 015015     045     116     045  XS0CON: .ASCIZ   '%N%A XSTAT0 Contents          = %06'
2758 015062     045     116     045  XS1CON: .ASCIZ   '%N%A XSTAT1 Contents          = %06'
2759 015127     045     116     045  XS2CON: .ASCIZ   '%N%A XSTAT2 Contents          = %06'
2760 015174     045     116     045  XS3CON: .ASCIZ   '%N%A XSTAT3 Contents          = %06'
2761                                  .EVEN
```

```
                                        .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
                                ;+
                                ;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
                                ;-
015242                          PRMSGEXP::
015242                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
015246  010005                          MOV     R0,R5           ;SAVE NUMBER OF WORDS
015250  013700  002252                  MOV     RCVLOADD,R0     ;GET RECV LOW ADDRESS
015254  010004                          MOV     R0,R4           ;COPY LOW ADDRESS
015256  013701  002250                  MOV     RCVHIADD,R1     ;GET RECV HIGH ADDRESS
015262  006100                          ROL     R0              ;SHIFT BIT15 TO C BIT
015264  006101                          ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
015266                                  PRINTX  #PRMSG0,R1,R4   ;PRINT MESSAGE BUFFER ADDRESS
015266  010446                          MOV     R4,-(SP)
015270  010146                          MOV     R1,-(SP)
015272  012746  015422                  MOV     #PRMSG0,-(SP)
015276  012746  000003                  MOV     #3,-(SP)
015302  010600                          MOV     SP,R0
015304  104415                          TRAP    C$PNTX
015306  062706  000010                  ADD     #10,SP
015312                                  PRINTX  #PRMSG1         ;PRINT HEADER FOR CONTENTS
015312  012746  015467                  MOV     #PRMSG1,-(SP)
015316  012746  000001                  MOV     #1,-(SP)
015322  010600                          MOV     SP,R0
015324  104415                          TRAP    C$PNTX
015326  062706  000004                  ADD     #4,SP
015332  005004                          CLR     R4              ;NUMBER OF THE CURRENT WORD
015334  012701  002266                  MOV     #EXPMSG,R1      ;GET EXPD BUFFER ADDRESS
015340  012702  002432                  MOV     #RECMSG,R2      ;GET RECV BUFFER ADDRESS
015344  011100                  20$:    MOV     (R1),R0         ;GET EXPD
015346  011203                          MOV     (R2),R3         ;GET RECV
015350                                  XOR     R0,R3           ;XOR EXPD/RECV
015360                                  PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
015360  010346                          MOV     R3,-(SP)
015362  012246                          MOV     (R2)+,-(SP)
015364  012146                          MOV     (R1)+,-(SP)
015366  010446                          MOV     R4,-(SP)
015370  012746  015525                  MOV     #PRMSG2,-(SP)
015374  012746  000005                  MOV     #5,-(SP)
015400  010600                          MOV     SP,R0
015402  104415                          TRAP    C$PNTX
015404  062706  000014                  ADD     #14,SP
015410  005204                          INC     R4              ;NUMBER OF THE NEXT
015412  020405                          CMP     R4,R5           ;DONE ALL YET?
015414  002001                          BGE     50$             ;BR IF YES
015416  000752                          BR      20$             ;DO ANOTHER
015420  000207                  50$:    RTS     PC              ;RETURN
015422  045  116  045          PRMSG0: .ASCIZ  '%N%A Message Buffer Address = %O1%O5'
015467  045  116  045          PRMSG1: .ASCIZ  '%N%A Message Buffer Contents:'
015525  045  116  045          PRMSG2: .ASCIZ  '%N%A   WORD #%D2%A  EXPD: %O6%A  RECV: %O6%A  XOR: %O6'
                                        .EVEN
```

```
                                        .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
                                ;-
2813 015612                     PRBYTEXP::
2814 015612                             SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
2815 015616  010005                     MOV     R0,R5                   ;SAVE NUMBER OF BYTES
2816 015620  005037  002264             CLR     PRMNO                   ;INIT ERROR COUNT
2817 015624  005004                     CLR     R4                      ;NUMBER OF THE CURRENT BYTE
2818 015626  012701  002266             MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
2819 015632  012702  002432             MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
2820 015636  111100             20$:    MOVB    (R1),R0                 ;GET EXPD BYTE
2821 015640  042700  177400             BIC     #^C<377>,R0             ;CLEAR UPPER BYTE
2822 015644  110037  016160             MOVB    R0,PRBEXP               ;SAVE FOR ERROR REPORT
2823 015650  111203                     MOVB    (R2),R3                 ;GET RECV BYTE
2824 015652  042703  177400             BIC     #^C<377>,R3             ;CLEAR UPPER BYTE
2825 015656  110337  016162             MOVB    R3,PRBREC               ;FOR ERROR REPORT
2826 015662                             XOR     R0,R3                   ;XOR EXPD/RECV
2827 015672  122122                     CMPB    (R1)+,(R2)+             ;EXPD = RECV?
2828 015674  001431                     BEQ     30$                     ;BR IF YES
2829 015676  005237  002264             INC     PRMNO                   ;UPDATE ERROR COUNT
2830 015702  023727  002264  000010     CMP     PRMNO,#8.               ;PRINTED 8?
2831 015710  101023                     BHI     30$                     ;BR IF YES
2832 015712                     27$:    PRINTX  #PRBMSG,R4,PRBEXP,PRBREC,R3
     015712  010346                     MOV     R3,-(SP)
     015714  013746  016162             MOV     PRBREC,-(SP)
     015720  013746  016160             MOV     PRBEXP,-(SP)
     015724  010446                     MOV     R4,-(SP)
     015726  012746  016026             MOV     #PRBMSG,-(SP)
     015732  012746  000005             MOV     #5,-(SP)
     015736  010600                     MOV     SP,R0
     015740  104415                     TRAP    C$PNTX
     015742  062706  000014             ADD     #14,SP
2833 015746                             FORCEXIT        50$             ;aad
2834 015756  000404                     BR      35$                     ;aad
2835 015760                     30$:
2836 015760                             FORCERROR  27$,NOTSSR           ;aad
2837 015770                     35$:                                    ;aad
2838 015770  005204                     INC     R4                      ;NUMBER OF THE NEXT
2839 015772  020405                     CMP     R4,R5                   ;DONE ALL YET?
2840 015774  002001                     BGE     50$                     ;BR IF YES
2841 015776  000717                     BR      20$                     ;DO ANOTHER
2842 016000                     50$:    PRINTX  #PRBTOT,PRMNO           ;PRINT TOTAL ERROR COUNT
     016000  013746  002264             MOV     PRMNO,-(SP)
     016004  012746  016113             MOV     #PRBTOT,-(SP)
     016010  012746  000002             MOV     #2,-(SP)
     016014  010600                     MOV     SP,R0
     016016  104415                     TRAP    C$PNTX
```

PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```
        016020  062706  000006                  ADD     #6,SP
   2843 016024  000207                          RTS     PC              ;RETURN
   2844
   2845 016026     045     116     045  PRBMSG: .ASCIZ  '%N%A   BYTE #%D2%A  EXPD: %O3%A  RECV. %O3%A  XOR: %O3'
   2846 016113     045     116     045  PRBTOT: .ASCIZ  '%N%A NUMBER OF BYTES IN ERROR = %D2'
   2847                                  .EVEN
   2848 016160  000000                  PPBEXP: .WORD   0               ;EXPD
   2849 016162  000000                  PRBREC: .WORD   0               ;RECV
   2850
   2851                                  ;+
   2852                                  ;
   2853                                  ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
   2854                                  ;
   2855                                  ;INPUTS:
   2856                                  ;
   2857                                  ;       R1      RECEIVED DATA
   2858                                  ;       R2      EXPECTED DATA
   2859                                  ;
   2860                                  ;-
   2861
   2862 016164                                  BGNMSG  EXPREC
        016164                           EXPREC::
   2863 016164  004737  007306                  JSR     PC,PRIXOR       ;PRINT THE DATA
   2864 016170                                  ENDMSG
        016170                           L10017:
        016170  104423                          TRAP    C$MSG
   2865
   2866
```

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

                                                BGNMSG  EXPBREC
                                        EXPBREC::
016172  004737  007156                          JSR     PC,PRIBXOR      ;PRINT THE DATA
016176                                          ENDMSG
                                        L10020:
016176  104423                                  TRAP    C$MSG


                                                .SBTTL  RAMERR  - PRINT RAM AND PACKET DATA
                                        ;+
                                        ;
                                        ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R4      POINTER TO COMMAND PACKET
                                        ;
                                        ;IMPL.  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
                                        ;-

                                                BGNMSG  RAMERR
                                        RAMERR::
016200  004737  013450                          JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
016204                                          ENDMSG
                                        L10021:
016204  104423                                  TRAP    C$MSG


                                                .SBTTL  RAMTADD - PRINT TEST ADDRESS, 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.
                                        ;       ERRHI           HIGH ORDER TEST ADDRESS
                                        ;       ERRLO           LOW ORDER TEST ADDRESS
                                        ;
                                        ;IMPLICIT OUTPUTS:
                                        ;
                                        ;       RAMSIZ  SET TO 0
                                        ;-
                                        ;
                                                BGNMSG  RAMTADD
                                        RAMTADD::
016206  004737  007640                          JSR     PC,PRITADD      ;PRINT TEST ADDRESS
016212  004737  013450                          JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                                ENDMSG
                                        L10022:
016216  104423                                 TRAP    C$MSG
                                        ;
                                        ;
                                                .SBTTL  RAMEXP  - PRINT RAM EXPD/RECV DATA
                                        ;+
                                        ;
                                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      RECEIVED DATA
                                        ;       R2      EXPECTED DATA
                                        ;       R4      CONTROLLER RAM ADDRESS
                                        ;-
                                        ;
                                                BGNMSG  RAMEXP
                                        RAMEXP::
016220  042701  177400                          BIC     #^C<377>,R1     ;SAVE EXPD RAM DATA BYTE
016224  042702  177400                          BIC     #^C<377>,R2     ;SAVE EXPD RAM DATA BYTE
016230  004737  007432                          JSR     PC,PRIRAM       ;PRINT THE RAM ADDRESS
016234  004737  007306                          JSR     PC,PRIXOR       ;PRINT THE DATA
                                                ENDMSG
                                        L10023:
016240  104423                                 TRAP    C$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
                                        TIMEXP::
                                                PRINTX  #TIMSG0         ;PRINT HEADER
016242  012746  016270                          MOV     #TIMSG0,-(SP)
016246  012746  000001                          MOV     #1,-(SP)
016252  010600                                  MOV     SP,R0
016254  104415                                  TRAP    C$PNTX
016256  062706  000004                          ADD     #4,SP
016262  004737  007306                          JSR     PC,PRIXOR       ;PRINT THE DATA
                                                ENDMSG
016266                                  L10024:
016266  104423                                  TRAP    C$MSG

016270    045     116     045   TIMSG0: .ASCIZ  '%N%A TIMER A STATUS IS IN BIT 3%N%A TIMER B STATUS IS IN BIT 2'
                                        .EVEN

                                        ;+
                                        ;
                                        ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      CONTENTS OF TSSR
                                        ;       R2      DATA WRITTEN (8 BITS)
                                        ;
                                        ;-
                                        ;
                                                BGNMSG  BADSSR
                                        BADSSR::
                                                MOV     R2,-(SP)        ;SAVE DATA TRANSFERRED
                                                BIC     #177400,R2      ;GET JUST ONE BYTE
                                                PRINTB  #XFERASC,R2
                                                MOV     R2,-(SP)
                                                MOV     #XFERASC,-(SP)
                                                MOV     #2,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTB
                                                ADD     #6,SP
                                                MOV     (SP)+,R2        ;RESTORE R2
                                                JSR     PC,PRITSSR      ;DECODE TSSR CONTENTS
                                                ENDMSG
                                        L10025:
                                                TRAP    C$MSG
                                        XFERASC:        .ASCIZ  '%N%A Data Transferred = %03'
```

```
                                        .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
                                ;
                                ;-

                                SOFINIT::
                                        SAVREG                  ; SAVE THE REGISTERS
016464 012765 000000 000000             MOV     #0,TSSR(R5)     ; DO THE INIT.
016470 004737 016740                    JSR     PC,WAITF        ; WAIT FOR SSR
016476 016500 000000                    MOV     TSSR(R5),R0     ;GET THE TSSR REGISTER
016502 010004                           MOV     R0,R4           ;TSSR CONTENTS
016506 042704 176277                    BIC     #^C<HIADDR!OFL>,R4
016510 052704 002200                    BIS     #SSR!NBA,R4     ;R4 HAS EXPECTED CONTENTS
016514 020400                           CMP     R4,R0           ;ONLY EXPECTED BITS SET ?
016520 001402                           BEQ     5$              ;BRANCH IF OKAY
016522 000241                           CLC                     ;CLEAR THE CARRY FOR ERROR
016524 000401                           BR      10$             ;GO TO EXIT
016526 000261                   5$:     SEC                     ;SET THE CARRY BIT
016530 000207                   10$:    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
                                ;
                                ;-

016534                          CHKAMB:
016534                                  SAVREG                  ;SAVE THE GENERAL REGISTERS
016540  010004                          MOV     R0,R4           ;CONTENTS OF TSSR
016542  032700  100000                  BIT     #SC,R0          ;IS BIT 15 SET ?
016546  001004                          BNE     5$              ;BRANCH IF YES
016550  032700  174077                  BIT     #^C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
016554  001023                          BNE     40$             ;MUST BE AN ERROR
016556  000424                          BR      45$             ;RETURN WITH SUCCESS
016560  032700  000200          5$:     BIT     #SSR,R0         ;IS READY BIT SET ?
016564  001011                          BNE     10$             ;BRANCH IF READY BIT IS SET.
016566  032700  000040                  BIT     #BIT5,R0        ;IS FATAL ERROR BIT SET ?
016572  001414                          BEQ     40$             ;ERROR IF NOT
016574  042704  177761                  BIC     #^CTERCLS,R4    ;CLEAR ALL BUT TERMINATION CODE
016600  020427  000016                  CMP     R4,#16          ;ALL THREE BITS MUST BE SET
016604  001007                          BNE     40$             ;ERROR IF NOT SET
016606  000410                          BR      45$             ;OK IF ALL ARE SET
016610  032700  000040          10$:    BIT     #BIT5,R0        ;IS FATAL ERROR BIT SET ?
016614  001405                          BEQ     45$             ;ERROR IF BIT IS SET WITH SSR
016616  032700  000006                  BIT     #BIT2!BIT1,R0   ;IS THIS A FUNCTION REJECT
016622  001002                          BNE     45$             ;BR, IF TSSR IS OK
016624  000241          40$:    CLC                             ;AMBIGUOUS CONTENTS
016626  000401                  BR      50$
016630  000261          45$:    SEC                             ;SHOW SUCCESS - NO AMBIGUITY
016632  000207          50$:    RTS     PC                      ;RETURN TO CALLER
```

```
                                        .SBTTL  ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
                                ;
                                ; DEFAULT DISPLAY INTERRUPT HANDLERS.
                                ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
                                ; OTHERWISE, SAVE CPU REGISTERS AND DISMISS.
                                ;

                                ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
                                ;
 000200                                 IOKCKIN=BIT7            ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
 000001                                 IOKSTP=BIT0             ; EXPECT "STOP" INTERRUPT.
                                ;
                                ;INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
016634   000                   INTMASK:        .BYTE   0
                                ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
016635   000                   INTFLAG:        .BYTE   0

                                ;SAVED INTERRUPT VECTOR:
016636  000000                  INTVEC: .WORD   0
                                ;SAVE CPU PC
016640  000000                  INTCPC: .WORD   0

                                ;SUBROUTINE TO ENABLE INTERRUPTS:
016642  010046                  ENAINT: MOV     R0,-(SP)        ;SAVE R0
016644  013700  002156                  MOV     IVEC,R0         ;GET POINTER TO VECTORS
016650  012720  016706                  MOV     #INTR,(R0)+     ;SET UP INTERRUPT VECTOR
016654  012720  000340                  MOV     #PRI07,(R0)+
016660  012600                          MOV     (SP)+,R0        ;RESTORE R0
016662  011646                          MOV     (SP),-(SP)
016664  012766  000000  000002          MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
016672  000002                          RTI

                                ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
016674  011646                  DSBINT: MOV     (SP),-(SP)
016676  012766  000340  000002          MOV     #PRI07,2(SP)
016704  000002                          RTI
```

```
                                                        .SBTTL  INTR    - INTERRUPT HANDLERS

                                                        BGNSRV  INTR            ;DEFINE INTERRUPT ENTRY
                                                INTR::
                                                        MOV     #1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
                                                        CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
                                                        BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
                                                        BNE     1$              ;BR IF YES
                                                        BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.

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

```
                                        .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
                                ;
016740                          WAITF:: BREAK                   ; DO A SUPVSR BREAK FIRST.
016740  104422                          TRAP    C$BRK
016742  012746  010000                  MOV     #10000,-(SP)    ;BIG MSEC TIMER
016746                                  DELAY   1               ;DELAY 100US
016746  012727  000001                  MOV     #1,(PC)+
016752  000000                          .WORD   0
016754  013727  002116                  MOV     L$DLY,(PC)+
016760  000000                          .WORD   0
016762  005367  177772                  DEC     -6(PC)
016766  001375                          BNE     .-4
016770  005367  177756                  DEC     -22(PC)
016774  001367                          BNE     .-20
016776  016500  000000          2$:     MOV     TSSR(R5),R0     ;READ THE TSSR REGISTER
017002  105700                          TSTB    R0              ;TEST FOR READY BIT SET

017004  100420                          BMI     3$              ; EXIT ON STOP FLAG.
017006                                  DELAY   1               ; WAIT 100 USEC
017006  012727  000001                  MOV     #1,(PC)+
017012  000000                          .WORD   0
017014  013727  002116                  MOV     L$DLY,(PC)+
017020  000000                          .WORD   0
017022  005367  177772                  DEC     -6(PC)
017026  001375                          BNE     .-4
017030  005367  177756                  DEC     -22(PC)
017034  001367                          BNE     .-20
017036  005316                          DEC     (SP)            ;REDUCE DELAY COUNT
017040  001356                          BNE     2$              ;RETRY UNTIL TIMER EXPIRES
017042  000241                          CLC                     ; C = 0, CONTROLLER STILL RUNNING...
017044  000401                          BR      4$              ;...OR HUNG-UP AFTER 300 MSEC.
017046  000261                  3$:     SEC                     ; C = 1, CONTROLLER IS STOPPED.
017050  005326                  4$:     DEC     (SP)+           ;RESTORE STACK WITHOUT CHANGING CARRY BIT
017052  000207                          RTS     PC
```

```
                        .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
                ;
                ;-

017054          CHKTSSR:
017054  004737  016740          JSR     PC,WAITF        ;WAIT FOR READY
017060  103014                  BCC     20$             ;BRANCH IF TIME OUT
017062  004737  016534          JSR     PC,CHKAMB       ;TSSR AMBIGOUS?
017066  103006                  BCC     10$             ;BR IF YES
017070  032700  100000          BIT     #SC,R0          ;SPECIAL CONDITION SET?
017074  001405                  BEQ     15$             ;BR IF NO
017076  032700  074000          BIT     #<SCE!BIE!RMR!NXM>,R0   ;ANY ERROR BITS SET?
017102  001402                  BEQ     15$             ;BR IF NO
017104  000241          10$:    CLC                     ;SET FAILURE
017106  000401                  BR      20$             ;
017110  000261          15$:    SEC                     ;SET SUCCESS
017112  000207          20$:    RTS     PC              ;RETURN TO CALLER
```

```
3207                                            .SBTTL  XNXM    - CHECK FOR NONEXISTENT MEMORY
3208                                    ;+
3209                                    ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
3210                                    ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
3211                                    ;               "C" = 0, ALL ADDRESSES OK.
3212                                    ;
3213                                    ;CALL:  MOV ADR1,R1
3214                                    ;       MOV ADR2,R2
3215                                    ;       JSR PC,NXM
3216                                    ;       RETURN                  ;TEST "C" AND PROCEED.
3217                                    ;
3218 017114  012737  017146  000004    XNXM:   MOV     #2$,@#4         ; SET BUSERR VECTOR.
3219 017122  012737  000200  000006            MOV     #PRI04,@#6
3220 017130  005003                            CLR     R3              ;FLAG.
3221 017132  005711                    1$:     TST     (R1)            ;TEST THE ADDRESS(ES).
3222                                                                    ;IF ANY TRAP, CONTINUE AT 2$.
3223 017134  020102                            CMP     R1,R2           ;OTHERWISE, CONTINUE HERE.
3224 017136  001407                            BEQ     3$              ;BR IF FINISHED (NO NEXM'S).
3225 017140  062701  000002                    ADD     #2,R1           ;SET NEXT ADDRESS...
3226 017144  000772                            BR      1$              ;...AND CONTINUE.
3227
3228 017146  005103                    2$:     COM     R3              ;GOT ONE, SET FLAG...
3229 017150  012716  017156                    MOV     #3$,(SP)
3230 017154  000002                            RTI                     ;...AND DISMISS INTERRUPT...
3231 017156                            3$:     CLRVEC  #4              ;...AND GIVE BACK THE VECTOR.
     017156  012700  000004                    MOV     #4,R0
     017162  104436                            TRAP    C$CVEC
3232 017164  005703                            TST     R3              ;DID WE CATCH ONE ??
3233 017166  001401                            BEQ     .+4             ;NO, "C" = 0, SKIP NEXT.
3234 017170  000261                            SEC                     ;YES, "C" = 1, (R1) = NEXM ADDR.
3235 017172  000207                            RTS     PC
3236
3237
3238
3239
3240                                            .SBTTL  TSTLOOP - CHECK ITERATION COUNT
3241                                    ;+
3242                                    ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
3243                                    ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
3244                                    ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
3245                                    ;
3246                                    ; CALL: LOOPTO  ARG
3247                                    ;
3248 017174                            TSTLOOP::
3249 017174  005737  002136                    TST     NOITS           ; ITERATIONS INHIBITED?
3250 017200  001006                            BNE     1$              ; YES.
3251 017202  005737  002152                    TST     QVP             ; NO.
3252 017206  100403                            BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
3253 017210  005337  002164                    DEC     LOOPCNT         ; BUMP LOOP COUNTER.
3254 017214  001002                            BNE     2$
3255 017216  000241                    1$:     CLC                     ;LOOP DISALLOWED, OR DONE.
3256 017220  000401                            BR      3$
3257 017222  000261                    2$:     SEC                     ;LOOP ENABLED.
3258 017224  000207                    3$:     RTS     PC
```

```
3260
3261
3262                                    .SBTTL  TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
3263                                ;+
3264                                ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
3265                                ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
3266                                ; IN THE CURRENT RUN SEQUENCE.
3267                                ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
3268                                ;
3269                                ;INPUT:
3270                                ;
3271                                ;       R0      POINTER TO TEST ID ASCIZ STRING
3272                                ;
3273                                ;OUTPUT:
3274                                ;
3275                                ;       R5      ADDRESS OF FIRST DEVICE REGISTER
3276                                ;
3277                                ;IMPLICIT OUTPUTS:
3278                                ;
3279                                ;       TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
3280                                ;
3281                                ;SIDE EFFECTS:
3282                                ;
3283                                ;       INTERRUPT LEVEL IS RASIED TO LEVEL OF
3284                                ;       THE DEVICE UNDER TEST
3285                                ;
3286                                ;-
3287
3288 017226                         TSTSETUP::
3289 017226  010046                        MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
3290 017230  005037  003106                CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
3291 017234  005037  017474                CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
3292 017240  005037  005232                CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
3293 017244  105037  016634                CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
3294 017250  013700  002150                MOV     UNITN,R0        ; GET THE UNIT NUMBER.
3295 017254  006300                        ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
3296 017256  005737  003062                TST     NODEV           ; DID STARTUP FIND THE DEVICE?
3297 017262  001430                        BEQ     4$              ; BR IF YES
3298 017264  100010                        BPL     3$              ; BR IF NOT IDLE
3299 017266  052760  160000  003130        BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
3300 017274                                ERRDF   1,NXR,NXRERR    ; NO DEVICE HERE -- PRINT IT
     017274  104455                        TRAP    C$ERDF
     017276  000001                        .WORD   1
     017300  003636                        .WORD   NXR
     017302  005176                        .WORD   NXRERR
3301 017304  000407                        BR      2$
3302 017306  052760  160001  003130  3$:   BIS     #160001,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
3303 017314                                ERRDF   2,NOINIT        ; DEVICE NOT IDLE
     017314  104455                        TRAP    C$ERDF
     017316  000002                        .WORD   2
     017320  004233                        .WORD   NOINIT
     017322  000000                        .WORD   0
3304 017324  012737  177777  003060  2$:   MOV     #-1,DUFLG       ; DROP THE UNIT
3305 017332                                DODU    UNITN
     017332  013700  002150                MOV     UNITN,R0
     017336  104451                        TRAP    C$DODU
3306 017340                                DOCLN                   ; ABORT THE PASS
```

```
                017340  104444                          TRAP    C$DCLN
3307            017342  000423                          BR      5$
3308
3309            017344                          4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
                017344  104421                          TRAP    C$RPLA
3310            017346  032700  001000                  BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
3311            017352  001412                          BEQ     1$              ; BR IF NO
3312            017354  011600                          MOV     (SP),R0         ;GET THE ID MESSAGE
3313            017356                                  PRINTF  #TNAM,R0        ;DISPLAY THE TEST ID
                017356  010046                          MOV     R0,-(SP)
                017360  012746  017422                  MOV     #TNAM,-(SP)
                017364  012746  000002                  MOV     #2,-(SP)
                017370  010600                          MOV     SP,R0
                017372  104417                          TRAP    C$PNTF
                017374  062706  000006                  ADD     #6,SP
3314            017400  005237  002162          1$:     INC     TSTCNT          ; BUMP TEST COUNTER.
3315            017404                                  SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
                017404  013700  002160                  MOV     IPRI,R0
                017410  104441                          TRAP    C$SPRI
3316            017412  005726                  5$:     TST     (SP)+           ;FIX UP THE STACK
3317            017414  013705  002154                  MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
3318            017420  000207                          RTS     PC
3319            017422     045     123     045  TNAM:  .ASCIZ  '%S%T%A Test'
3320                                                    .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).
                                        ;
017436                                  TSTEND: RFLAGS  R0
017436  104421                                 TRAP    C$RFLA
017440  030027  020000                          BIT     R0,#IER
017444  001412                                 BEQ     1$              ; BR IF "IER" NOT SET.
017446                                          PRINTF  #ESUM,ERRK      ; PRINT ERROR COUNT.
017446  013746  017474                         MOV     ERRK,-(SP)
017452  012746  017476                         MOV     #ESUM,-(SP)
017456  012746  000002                         MOV     #2,-(SP)
017462  010600                                 MOV     SP,R0
017464  104417                                 TRAP    C$PNTF
017466  062706  000006                         ADD     #6,SP
017472  000207                          1$:     RTS     PC

017474  000000                          ERRK:   0                       ; LOCAL ERROR COUNT.
017476     045     101     040          ESUM:   .ASCIZ  /%A %D%A ERRORS/
017515     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:
                                        ;-
017562  005237  017474                  INCERK: INC     ERRK            ; INCREMENT LOCAL ERROR COUNT
017566  010046                                  MOV     R0,-(SP)        ; SAVE R0
017570  013700  002150                          MOV     UNITN,R0        ; GET UNIT NUMBER,
017574  006300                                  ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
017576  062700  003130                          ADD     #ERTABL,R0      ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
017602  005210                                  INC     (R0)            ; INCREMENT THE DEVICE ERROR COUNT
017604  032710  007777                          BIT     #7777,(R0)      ; DID WE OVERFLOW THE FIELD?
017610  001001                                  BNE     1$              ; BR IF NO.
017612  005310                                  DEC     (R0)            ; YES -- BACK IT UP TO 7777.
017614  012600                          1$:     MOV     (SP)+,R0        ; RESTORE R0
017616  000207                                  RTS     PC              ; RETURN TO CALLER.

017620  010046                          CKEMAX: MOV     R0,-(SP)        ; SAVE R0
017622  013700  002150                          MOV     UNITN,R0        ; GET UNIT NUMBER
017626  006300                                  ASL     R0              ; ... AND MAKE IT A WORD OFFSET
017630  016000  003130                          MOV     ERTABL(R0),R0   ; GET ERROR TABLE ENTRY
017634  042700  170000                          BIC     #170000,R0      ; EXTRACT ERROR COUNT FIELD
017640  020037  002142                          CMP     R0,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
017644  103004                                  BHIS    1$              ; BR IF YES
017646  023737  017474  002140                  CMP     ERRK,LERRMAX    ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
017654  103417                                  BLO     2$              ; BR IF NO
017656                                  1$:     RFLAGS  R0              ; GET OPERATOR FLAGS
017656  104421                                  TRAP    C$RFLA
017660  032700  000040                          BIT     #IDU,R0         ; IS DROPPING INHIBITED?
017664  001013                                  BNE     2$              ; BR IF YES.
017666  012737  177777  003060                  MOV     #-1,DUFLG       ; NO -- DROP THE UNIT
017674                                          ERRDF   4,EMAXDU
017674  104455                                  TRAP    C$ERDF
017676  000004                                  .WORD   4
017700  017515                                  .WORD   EMAXDU
```

```
                017702  000000                          .WORD   0
     3368       017704                                  DODU    UNITN
                017704  013700  002150                  MOV     UNITN,R0
                017710  104451                          TRAP    C$DODU
     3369       017712                                  DOCLN
                017712  104444                          TRAP    C$DCLN
     3370       017714  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
     3371       017716  000207                          RTS     PC              ; RETURN TO CALLER
     3372                                               .SBTTL  FATCHK - INC FATAL ERRORS AND CHECK FOR LIMIT
     3373                                           ;+
     3374                                           ;
     3375                                           ;       CHECK FATAL COUNTER, AFTER INC, FOR MORE THAN 25
     3376                                           ;       ERRORS AND IF OVER CALL UNIT DROP ROUTINE
     3377                                           ;
     3378                                           ;-
     3379       017720                              FATCHK:
     3380       017720                                      SAVREG                  ;BETTER SAVE THE REGISTERS
     3381       017724  013701  002150                      MOV     UNITN,R1        ;PICK UP THE UNIT NUMBER
     3382       017730  006301                              ASL     R1              ;MAKE IT INTO A BYTE OFFSET
     3383       017732  062761  000001  003130              ADD     #1,ERTABL(R1)   ;ADD 1 TO THE PROPER UNIT'S ERROR COUNTER
     3384       017740  005237  002170                      INC     FATFLG          ;BUMP FATAL ERROR COUNTER
     3385       017744  023727  002170  000031              CMP     FATFLG,#25.     ;CHECK AGAINST 25
     3386       017752  002406                              BLT     9$              ;BR. IF LESS THAN 25 ERRORS
     3387       017754                                      RFLAGS  R0              ;READ THE FLAGS INTO R0
                017754  104421                              TRAP    C$RFLA
     3388       017756  032700  040000                      BIT     #BIT14,R0       ;BR. IF LOOP ON ERROR IS SET
     3389       017762  001002                              BNE     9$              ;OTHERWISE NEVER BE ABLE TO SCOPE ETC.
     3390       017764  004737  017772                      JSR     PC,CKDROP       ;DROP UNIT IF ALLOWED
     3391       017770  000207                  9$:     RTS     PC              ;RETURN ETC.
     3392                                           ;
     3393                                           ;
     3394                                           ;
```

```
                                                        .SBTTL  CKDROP  - CHECK IF UNIT SHOULD BE DROPPED
                                                ;+
                                                ; CHECK IF UNIT SHOULD BE DROPPED
                                                ;-
020000 017772  010046                           CKDROP: MOV     R0,-(SP)
020002 017774                                           FORCERROR       1$,NOTSSR
020004 020004                                           RFLAGS  R0
       020004  104421                                   TRAP    C$RFLA
020006 020006  032700  000040                           BIT     #IDU,R0
020012 020012  001010                                   BNE     1$
020014 020014  011600                                   MOV     (SP),R0
020016 020016  012737  177777  003060                   MOV     #-1,DUFLG
020024 020024                                           DODU    UNITN
       020024  013700  002150                           MOV     UNITN,R0
       020030  104451                                   TRAP    C$DODU
020032 020032                                           DOCLN                   ;ABORT THE PASS
       020032  104444                                   TRAP    C$DCLN
020034 020034  012600                           1$:     MOV     (SP)+,R0
020036 020036  000207                                   RTS     PC


                                                        .SBTTL  CONFIG  - DETERMINE CONFIGURATION OF SYSTEM
                                                ;
                                                ; SUBROUTINE - DETERMINE CONFIGURATION OF TU80 SYSTEM.
                                                ;
020040 020040                                   CONFIG:
020040 020040  004737  016464                           JSR     PC,SOFINIT
020044 020044  000207                                   RTS     PC

                                                ;
                                                ; SUBROUTINE - ENABLE MEM MGT.
                                                ;
020046 020046  005737  003100                   KTON:   TST     KTFLG           ; GOT KT?
020052 020052  001403                                   BEQ     1$              ; NO.
020054 020054  012737  000001  177572                   MOV     #1,SR0          ; YES. ENABLE KT11.
020062 020062  000207                           1$:     RTS     PC

                                                ;
                                                ; SUBROUTINE - DISABLE MEM MGT.
                                                ;
020064 020064  005737  003100                   KTOFF:  TST     KTFLG           ; GOT KT11?
020070 020070  001405                                   BEQ     1$              ; NO.
020072 020072  000240                                   NOP
020074 020074  000240                                   NOP
020076 020076  012737  000000  177572                   MOV     #0,SR0          ; DISABLE KT.
020104 020104  000207                           1$:     RTS     PC
```

```
                        .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
                ;-
020106          SETMAP:
020106                  SAVREG                          ;SAVE R1-R4 UNTIL NEXT RETURN
020112  005737  003100          TST     KTFLG           ;SYSTEM HAVE ABOVE 28K?
020116  001433                  BEQ     10$             ;BR IF NO
020120  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
020152  042701  000177          BIC     #177,R1         ;ALINE FOR LOWER 4K BOUNDARY
020156  020137  003100          CMP     R1,KTFLG        ;HIGHER THAN EXISTING MEMORY?
020162  103011                  BHIS    10$             ;BR IF YES
020164  010137  172354          MOV     R1,@#K1PAR6     ;SETUP MAPPING REGISTER PAR6
020170  042702  160000          BIC     #160000,R2      ;SETUP DISPLACEMENT IN PAGE
020174  062702  140000          ADD     #140000,R2      ;ADD IN PAR6 BIAS
020200  010200                  MOV     R2,R0           ;RETURN IN R0
020202  000261                  SEC                     ;SET SUCCESS
020204  000401                  BR      15$             ;
020206  000241          10$:    CLC                     ;SET FAILURE
020210  000207          15$:    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
                                ;-
                                ;
020212                          FILLMEM:
020212                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
020216  004737  020064                  JSR     PC,KTOFF        ;DISABLE KT.
020222  010003                          MOV     R0,R3           ;COPY TEST PATTERN
020224  013701  003072                  MOV     FREE,R1         ;GET FIRST FREE LOCATION
020230  013702  003074                  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
020234  010321                  10$:    MOV     R3,(R1)+        ;STORE A BACKGROUND WORD
020236  005302                          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
020240  003375                          BGT     10$             ;BR IF NO
020242  005737  003100                  TST     KTFLG           ; GOT KT?
020246  001452                          BEQ     55$             ; NO. GET OUT.
020250  004737  020046                  JSR     PC,KTON         ; YES. ENABLE KT.
020254  005000                          CLR     R0              ;HIGH ORDER ADDRESS START
020256  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
020326  004737  020106                  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
020332  010320                  30$:    MOV     R3,(R0)+        ;STORE TEST PATTERN IN >28K ADDRESS
020334  020027  160000                  CMP     R0,#160000      ;END OF PAR6 MAPPING AREA?
020340  103774                          BLO     30$             ;BR IF NO
020342  162700  020000                  SUB     #20000,R0       ;BACKUP INTO PAR6 MAPPING BEGIN
020346  062737  000200  172354          ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
020354  023737  172354  003100          CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
020362  001402                          BEQ     50$             ;BR IF YES
020364  000137  020332                  JMP     30$             ;KEEP GOING ON ETC.
020370  004737  020064          50$:    JSR     PC,KTOFF        ; DISABLE KT.
020374  000207                  55$:    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
                                ;-
020376                          CMPMEM:
020376                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
020402  010003                          MOV     R0,R3           ;COPY TEST PATTERN
020404  004737  020064                  JSR     PC,KTOFF        ;DISABLE KT.
020410  013701  003072                  MOV     FREE,R1         ;GET FIRST FREE LOCATION
020414  013702  003074                  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
020420  020311                  10$:    CMP     R3,(R1)         ;FREE SPACE LOCATION EQUAL TO EXPD?
020422  001411                          BEQ     15$             ;BR IF YES
020424  010137  002204                  MOV     R1,ERRLO        ;SAVE ADDRESS IN ERROR
020430  005037  002202                  CLR     ERRHI           ;NO HIGH ADDRESS
020434  010337  002176                  MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
020440  011137  002200                  MOV     (R1),RECV       ;SAVE RECV FOR ERROR REPORT
020444  000474                          BR      50$             ;
020446  005721                  15$:    TST     (R1)+           ;POINT TO NEXT ADDRESS
020450  005302                          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
020452  003362                          BGT     10$             ;BR IF NO
020454  005737  003100                  TST     KTFLG           ; GOT KT?
020460  001472                          BEQ     55$             ; NO. GET OUT.
020462  004737  020046                  JSR     PC,KTON         ; YES. ENABLE KT.
020466  005000                          CLR     R0              ;HIGH ORDER ADDRESS START
020470  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
020524  042701  000177                  BIC     #177,R1         ;ALINE 4K BOUNDARY
020530  010046                          MOV     R0,-(SP)        ;SAVE HIGH ORDER
020532  010146                          MOV     R1,-(SP)        ;SAVE LOW ORDER
020534  004737  020106                  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
020540  010004                          MOV     R0,R4           ;COPY ADDRESS BIASED TO PAR6
020542  012601                          MOV     (SP)+,R1        ;RESTORE LOW ORDER IN NON PAR6 FORMAT
020544  012600                          MOV     (SP)+,R0        ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
020546  020314                  30$:    CMP     R3,(R4)         ;ABOVE 28K LOCATION EQUAL EXPD?
020550  001411                          BEQ     32$             ;BR IF YES
020552  010037  002202                  MOV     R0,ERRHI        ;SAVE HIGH ORDER IN ERROR
```

```
3587 020556  010137  002204                    MOV     R1,ERRLO        ;SAVE LOW ORDER IN ERROR
3588 020562  010337  002176                    MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
3589 020566  011437  002200                    MOV     (R4),RECV       ;SAVE RECV FOR ERROR REPORT
3590 020572  000421                            BR      50$             ;
3591 020574  062701  000002          32$:      ADD     #2,R1           ;UPDATE NON PAR6 ADDRESS
3592 020600  005500                            ADC     R0              ;MAKE IT DOUBLE PRECISION ADD
3593 020602  062704  000002                    ADD     #2,R4           ;UPDATE PAR FORMAT ADDRESS
3594 020606  020427  160000                    CMP     R4,#160000      ;END OF PAR6 MAPPING AREA?
3595 020612  103755                            BLO     30$             ;BR IF NO
3596 020614  162704  020000                    SUB     #20000,R4       ;BACKUP INTO PAR6 MAPPING BEGIN
3597 020620  062737  000200  172354            ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
3598 020626  023737  172354  003100            CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
3599 020634  101744                            BLOS    30$             ;BR IF NO
3600 020636  004737  020064          50$:      JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
3601 020642  000241                            CLC                     ;SET FAILURE
3602 020644  000403                            BR      60$             ;
3603 020646  004737  020064          55$:      JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
3604 020652  000261                            SEC                     ;SET SUCCESS
3605 020654  000207          60$:      RTS     PC
3606
```

```
                                        .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
                                ;
                                ;-

020656                          REGSAV:
020656                                  BREAK                   ;LOOK FOR CNTL C
020656  104422                          TRAP    C$BRK
020660  010446                          MOV     R4,-(SP)
020662  010346                          MOV     R3,-(SP)
020664  010246                          MOV     R2,-(SP)
020666  010146                          MOV     R1,-(SP)
020670  010546                          MOV     R5,-(SP)
020672  016605  000012                  MOV     10.(SP),R5
020676  004736                          JSR     PC,@(SP)+
020700  012601                          MOV     (SP)+,R1
020702  012602                          MOV     (SP)+,R2
020704  012603                          MOV     (SP)+,R3
020706  012604                          MOV     (SP)+,R4
020710  012605                          MOV     (SP)+,R5
020712                                  BREAK                   ;LOOK FOR CNTL C
020712  104422                          TRAP    C$BRK
020714  000207                          RTS     PC
```

```
                                        .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
                                1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
020722  104443                          TRAP    C$GMAN
020724  000406                          BR      10000$
020726  020752                          .WORD   PATDAT
020730  000022                          .WORD   T$CODE
020732  020754                          .WORD   DATASC
020734  000377                          .WORD   377
020736  000000                          .WORD   T$LOLIM
020740  000377                          .WORD   T$HILIM
020742                          10000$:
020742                                  BNCOMPLETE      1$      ;RETRY IF ERROR
020742  103367                          BCC     1$
020744  013700  020752                  MOV     PATDAT,R0       ;DATA PATTERN FROM OPERATOR
020750  000207                          RTS     PC              ;RETURN TO CALLER

                                ;+
                                ;LOCAL DATA AREA
                                ;-

020752  000000                  PATDAT: .WORD   0               ;TEMPORARY STORAGE FOR DATA
020754     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
                                ;-
021000                          GETSEL::
021000                                  SAVREG                          ;SAVE GENERAL REGIS ERS
021004  010002                          MOV     R0,R2                   ;SAVE THE MENU ADDRESS
021006  010203                  1$:     MOV     R2,R3                   ;START OF MENU STRING
021010  005713                  2$:     TST     (R3)                    ;END OF ASCII ?
021012  001412                          BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
021014                                  PRINTF  #SELASC,(R3)+           ;DISPLAY THE MENU
021014  012346                          MOV     (R3)+,-(SP)
021016  012746  021164                  MOV     #SELASC,-(SP)
021022  012746  000002                  MOV     #2,-(SP)
021026  010600                          MOV     SP,R0
021030  104417                          TRAP    C$PNTF
021032  062706  000006                  ADD     #6,SP
021036  000764                          BR      2$
021040                          3$:     GMANID  MENASC,MENRES,D,-1,0,-1,NO
021040  104443                          TRAP    C$GMAN
021042  000406                          BR      10001$
021044  021220                          .WORD   MENRES
021046  000042                          .WORD   T$CODE
021050  021171                          .WORD   MENASC
021052  177777                          .WORD   -1
021054  000000                          .WORD   T$LOLIM
021056  177777                          .WORD   T$HILIM
021060                          10001$:
021060                                  BNCOMPLETE      1$              ;RETRY IF ERROR
021060  103352                          BCC     1$
021062  013700  021220                  MOV     MENRES,R0               ;GET THE OPERATOR'S REPLY
021066  020001                          CMP     R0,R1                   ;COMPARE TO MAXIMUM ALLOWED
021070  101411                          BLOS    5$                      ;BRANCH IF OK
021072                                  PRINTF  #MENERR                 ;DISPLAY ERROR MESSAGE
021072  012746  021116                  MOV     #MENERR,-(SP)
021076  012746  000001                  MOV     #1,-(SP)
021102  010600                          MOV     SP,R0
021104  104417                          TRAP    C$PNTF
021106  062706  000004                  ADD     #4,SP
021112  000735                          BR      1$                      ;RETRY
021114  000207                  5$:     RTS     PC                      ;RETURN TO CALLER
021116     045     116     045  MENERR: .ASCIZ   '%N%A *** Menu Selection Too Large ***'
021164     045     116     045  SELASC: .ASCIZ   '%N%T'
021171     105     156     164  MENASC: .ASCIZ   'Enter Menu Selection: '
                                        .EVEN
021220  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.
                                ;
                                ;-

021222                          CHKMAN::
021222                                  SAVREG                  ;SAVE THE REGISTERS
021226                                  MANUAL                  ;SEE IF MANUAL INTERVENTION OK
021226  104450                          TRAP    C$MANI
021230                                  BCOMPLETE 1$            ;BRANCH IF ALLOWED
021230  103411                          BCS     1$
021232                                  PRINTF  #NOMAN          ;PRINT THE WARNING MESSAGE
021232  012746  021256                  MOV     #NOMAN,-(SP)
021236  012746  000001                  MOV     #1,-(SP)
021242  010600                          MOV     SP,R0
021244  104417                          TRAP    C$PNTF
021246  062706  000004                  ADD     #4,SP
021252  000241                          CLC                     ;CLEAR CARRY FOR ERROR
021254  000207                  1$:     RTS     PC              ;RETURN

021256      045     116     045 NOMAN:  .ASCIZ  '%N%A *** Manual Intervention not Allowed - Test Aborted ***'
                                        .even
```

```
                                                .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
                                        ;
                                        ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
                                        ;
021352                                  ENVIRN: MEMORY  R0
021352  104431                                  TRAP    C$MEM
021354  010037  003072                          MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
021360  062737  000002  003072                  ADD     #2,FREE
021366  011037  003074                          MOV     (R0),FRESIZ     ;...AND WORD COUNT.
021372  162737  000004  003074                  SUB     #4,FRESIZ
021400  013702  002012                          MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
021404  162737  000007  003074  10$:            SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
021412  005302                                  DEC     R2
021414  001373                                  BNE     10$
021416  013700  003072                          MOV     FREE,R0         ;GET FIRST FREE ADDRESS
021422  063700  003074                          ADD     FRESIZ,R0       ;POINT TO LAST FREE ADDRESS
021426  162700  000002                          SUB     #2,R0           ;BACKUP 1 WORD
021432  010037  003076                          MOV     R0,FREEHI       ;STORE LAST FREE ADDRESS
021436  000207                                  RTS     PC              ;RETURN
```

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

021440                          KTINIT
021440  005037  003100                  CLR     KTFLG           ; INIT >28K MEMORY FLAG
021444  005037  003102                  CLR     KTENABLE        ; INIT TEST >28K FLAG
021450  023727  002120  001577          CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
021456  101444                          BLOS    9$              ; NO.
021460  013700  000004                  MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
021464  012737  021556  000004          MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
021472  005737  177572                  TST     @#SR0           ; GOT KT11?
021476  000240                          NOP                     ;  (TRAP IF NO).
021500  013737  002120  003100          MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
021506  042737  000177  003100          BIC     #177,KTFLG      ;
021514  010037  000004                  MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
021520  005000                          CLR     R0              ; R0 = AR DATA.
021522  012701  172340                  MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
021526  012761  077406  177740  1$:     MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
021534  010021                          MOV     R0,(R1)+        ; SET KIPAR REG.
021536  062700  000200                  ADD     #200,R0         ; BUMP AR DATA BY "4K".
021542  020027  002000                  CMP     R0,#2000        ; AT "I/O"?
021546  001367                          BNE     1$              ; NO.
021550  012741  177600                  MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
021554  000405                          BR      9$

021556  012716  021564          2$:     MOV     #6$,(SP)        ; SET UP RETURN
021562  000002                          RTI                     ; RTI TO NEXT LOCATION

021564  010037  000004          6$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.

021570  000207                  9$:     RTS     PC
                                ;       .IIF DF ONEFILE,        .PAGE
                                                        .SBTTL  PROTECTION TABLE
021572                                  BGNPROT
021572                          L$PROT::
021572  177777  177777  177777          .WORD   -1, -1, -1, -1  ;NO DEVICE PROTECTION REQUIRED.
021602                                  ENDPROT
```

```
                                        .SBTTL  'NITIALIZE 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
                                ;-
021602                                  BGNINIT
021602                          L$INIT::
021602                          40$:
021602  012737  005672  002146         MOV     #EPRT1,EPRTSW   ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021610  005037  003106                 CLR     SIFLAG          ;CLEAR "SOFT INIT" FLAG
021614  005037  003102                 CLR     KTENABLE        ;CLEAR TEST ABOVE 28K FLAG
021620  005037  002246                 CLR     RAMSIZ          ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021624                                  READEF  #EF.CONTINUE
021624  012700  000036                 MOV     #EF.CONTINUE,R0
021630  104447                         TRAP    C$REFG
021632                                  BNCOMPLETE 1$
021632  103023                         BCC     1$
021634  023737  002150  002012         CMP     UNITN,L$UNIT    ;UNIT IN RANGE?
021642  103073                         BHIS    4$              ;BR IF NO.
021644  005737  003060                 TST     DUFLG           ;DROPPED UNIT?
021650  100475                         BMI     NXTU            ;BR IF YES
021652  013701  002150                 MOV     UNITN,R1
021656  006301                         ASL     R1
021660  005761  003130                 TST     ERTABL(R1)
021664  001521                         BEQ     SETU
021666  032761  040000  003130         BIT     #BIT14,ERTABL(R1) ;DROPPED?
021674  001063                         BNE     NXTU
021676                                  EXIT    INIT            ;DO NOTHING IF "CONTINUE".
021676  104432                         TRAP    C$EXIT
021700  000430                         .WORD   L10030-.
021702                          1$:     READEF  #EF.NEW
021702  012700  000035                 MOV     #EF.NEW,R0
021706  104447                         TRAP    C$REFG
021710                                  BNCOMPLETE NXTU         ;TAKE NEXT UNIT IF NOT NEW PASS.
021710  103055                         BCC     NXTU
021712                                  READEF  #EF.START
021712  012700  000040                 MOV     #EF.START,R0
021716  104447                         TRAP    C$REFG
021720                                  BCOMPLETE 2$
021720  103404                         BCS     2$
021722                                  READEF  #EF.RESTART
021722  012700  000037                 MOV     #EF.RESTART,R0
021726  104447                         TRAP    C$REFG
021730                                  BNCOMPLETE 31$
021730  103034                         BCC     31$
021732                          2$:                             ;1ST PASS, BUS-INIT...
021732                                  BRESET                  ;BUS RESET.
021732  104433                         TRAP    C$RESET
021734  005037  002162                 CLR     TSTCNT          ;NUMBER OF TESTS RUN IN PASS
```

```
3864 021740  005037  002170                CLR     FATFLG          ;RESET FLAG TO ZERO "FATAL ERRORS"
3865 021744  000406                        BR      19$             ;BR, IF THE FLAG IS NOT SET
3866                                                               ;(NO DEBUGGER ETC.)
3867 021746  012746  000340                MOV     #340,-(SP)
3868 021752  012746  021766                MOV     #20$,-(SP)      ;RETURN TO DEBUGGER
3869 021756  000137  046040                JMP     O.ODT           ;;@@ENTER THE DEBUGGER
3870 021762  005037  003332          19$:  CLR     SKIPT           ;CLEAR THE SUBTEST "SKIPPER"
3871 021766                          20$:
3872 021766  012737  177777  002152        MOV     #-1,QVP         ;...QUICK VERIFY...
3873 021774  004737  021352                JSR     PC,ENVIRN       ;SET ENVIRONMENT.
3874 022000  004737  021440                JSR     PC,KTINIT       ;INITIALIZE KT MEMORY MANAGEMENT
3875 022004  012700  003130                MOV     #ERTABL,R0
3876 022010  005020                  30$:  CLR     (R0)+           ;CLEAR THE ERROR TABLE
3877 022012  020027  003330                CMP     R0,#ERTABE
3878 022016  103774                        BLO     30$
3879 022020  000404                        BR      4$
3880 022022  005037  002152          31$:  CLR     QVP
3881 022026  000137  022076                JMP     PASRPT          ;GO REPORT THE STATUS
3882
3883 022032                          4$:
3884 022032  012737  177777  002150  NEWPAS: MOV     #-1,UNITN       ;INIT UNIT NUMBER...
3885 022040  005037  002166                CLR     DEVCNT          ;CLEAR COUNT OF DEVICES RUNNING
3886 022044                          NXTU:   BREAK
     022044  104422                        TRAP    C$BRK
3887 022046  005237  002150                INC     UNITN           ;...AND SET NEXT UNIT NUMBER.
3888 022052  023737  002150  002012        CMP     UNITN,LSUNIT
3889 022060  103423                        BLO     SETU
3890 022062  012737  177777  003060        MOV     #-1,DUFLG
3891 022070  000401                        BR      11$
3892 022072                                DDCLN                   ;ABORT, NO MORE UNITS.
     022072  104444                        TRAP    C$DCLN
3893 022074  000240                  11$:  NOP
3894 022076                          PASRPT:
3895 022076  023727  002012  000001        CMP     L$UNIT,#1       ;HOW MANY UNITS SELECTED?
3896 022104  101752                        BLOS    NEWPAS          ;BR IF ONLY 1
3897 022106  005737  002166                TST     DEVCNT          ;ARE ANY STILL RUNNING?
3898 022112  001747                        BEQ     NEWPAS          ;BR IF NO
3899 022114                                RFLAGS  R0
     022114  104421                        TRAP    C$RFLA
3900 022116  032700  000100                BIT     #ISR,R0         ;SHOULD WE PRINT STATISTICS
3901 022122  001343                        BNE     NEWPAS          ;BR IF NO
3902
3903 022124                                DORPT
     022124  104424                        TRAP    C$DRPT
3904 022126  000741                        BR      NEWPAS
3905 022130                          10$:
3906
3907 022130                          SETU:   GPHARD  UNITN,R0        ;GET UNIT N P-TABLE POINTER.
     022130  013700  002150                MOV     UNITN,R0
     022134  104442                        TRAP    C$GPHRD
3908 022136                                BNCOMPLETE NXTU         ;BR IF UNIT NOT AVAILABLE.
     022136  103342                        BCC     NXTU
3909 022140  005037  003060                CLR     DUFLG           ;CLEAR "DROPPED" FLAG.
3910 022144  005237  002166                INC     DEVCNT
3911 022150  012001                        MOV     (R0)+,R1        ;GET 1ST REGISTER ADDRESS.
3912 022152  010137  002154                MOV     R1,CSRADDR      ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
3913
```

```
3914 022156  012001                          MOV     (R0)+,R1        ;GET VECTOR ADDRESS.
3915 022160  011002                          MOV     (R0),R2         ;GET INTERRUPT PRIORITY
3916 022162  010237  002160                  MOV     R2,IPRI         ;SET INTERRUPT PRIORITY.
3917 022166  010137  002156                  MOV     R1,IVEC         ;SET INTERRUPT VECTOR POINTER...
3918 022172  012721  016706                  MOV     #INTR,(R1)+     ;...VECTOR...
3919 022176  010221                          MOV     R2,(R1)+        ;...AND PRIORITY.
3920
3921 022200                          1$:
3922                                 ;       TST     QVP             ;1ST PASS ??
3923                                 ;       BEQ     5$              ;NO, SKIP THE PASS 1 STUFF.
3924
3925                                 ;
3926                                 ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
3927                                 ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
3928                                 ;
3929 022200  013701  002150                  MOV     UNITN,R1
3930 022204  006301                          ASL     R1
3931 022206  052761  100000  003130          BIS     #BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
3932 022214  005037  005232                  CLR     EXTA            ;CLEAR ERROR EXTENSION FLAG.
3933 022220  023727  002012  000001          CMP     LSUNIT,#1       ;ARE WE TESTING MULTIPLE UNITS?
3934 022226  101416                          BLOS    10$             ;BR IF NO.
3935 022230                                  RFLAGS  R0              ;YES -- GET OPERATOR FLAGS.
     022230  104421                          TRAP    C$RFLA
3936 022232  032700  001000                  BIT     #PNT,R0         ;SHOULD WE PRINT UNIT #?
3937 022236  001412                          BEQ     10$             ;BR IF NOT.
3938 022240                                  PRINTF  #PUNIT,UNITN    ;PRINT THE UNIT #
     022240  013746  002150                  MOV     UNITN,-(SP)
     022244  012746  022332                  MOV     #PUNIT,-(SP)
     022250  012746  000002                  MOV     #2,-(SP)
     022254  010600                          MOV     SP,R0
     022256  104417                          TRAP    C$PNTF
     022260  062706  000006                  ADD     #6,SP
3939 022264                          10$:
3940 022264  005037  003062                  CLR     NODEV
3941 022270  013701  002154                  MOV     CSRADDR,R1      ;ADDRESS OF FIRST REGISTER
3942 022274  010102                          MOV     R1,R2           ;START OF REGISTERS
3943 022276  062702  000000                  ADD     #TSSR,R2        ;ADDRESS OF TSSR REGISTER
3944 022302  004737  017114                  JSR     PC,XNXM         ;TEST BOTH CONTROLLER REGISTERS...
3945 022306  103005                          BCC     2$              ;...AND BR IF ALL OK.
3946 022310  010137  003062                  MOV     R1,NODEV        ;FLAG DEVICE AS NON-EXISTENT
3947 022314  012737  177777  003060          MOV     #-1,DUFLG       ;DROP THIS UNIT.
3948 022322                          2$:
3949                                 ;
3950                                 ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
3951                                 ;
3952 022322                          5$:     SETPRI  #PRI00          ;ENABLE INTERRUPTS.
     022322  012700  000000                  MOV     #PRI00,R0
     022326  104441                          TRAP    C$SPRI
3953 022330                                  ENDINIT
     022330                          L10030:
     022330  104411                          TRAP    C$INIT
3954
3955 022332     045     116     045  PUNIT:  .ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
3956                                          .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.
                                        ;--
022400                                          BGNAU
022400                                  L$AU::
022400  010001                                  MOV     R0,R1           ; GET UNIT TO BE ADDED (R0)
022402  006301                                  ASL     R1              ; MAKE IT A WORD INDEX
022404  052761  100000  003130                  BIS     #100000,ERTABL(R1) ; SET THE "ACTIVE" BIT
022412  042761  040000  003130                  BIC     #40000,ERTABL(R1)  ; CLEAR THE "DROPPED" BIT
022420                                          PRINTF  #1$,R0
022420  010046                                  MOV     R0,-(SP)
022422  012746  022446                          MOV     #1$,-(SP)
022426  012746  000002                          MOV     #2,-(SP)
022432  010600                                  MOV     SP,R0
022434  104417                                  TRAP    C$PNTF
022436  062706  000006                          ADD     #6,SP
022442                                          EXIT    AU
022442  000167                                  .WORD   J$JMP
022444  000026                                  .WORD   L10031-2-.
022446    045     116     045   1$:             .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                                .EVEN

022474                                          ENDAU                   ; UNUSED.
022474                                  L10031:
022474  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.

022476                                          BGNDU
022476                                  L$DU::
022476  012737  177777  003060                  MOV     #-1,DUFLG
022504  010001                                  MOV     R0,R1
022506  006301                                  ASL     R1
022510  052761  140000  003130                  BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022516  000240  000240  000240                  240,240,240                     ; ?????????
022524                                          PRINTF  #1$,R0
022524  010046                                  MOV     R0,-(SP)
022526  012746  022552                          MOV     #1$,-(SP)
022532  012746  000002                          MOV     #2,-(SP)
022536  010600                                  MOV     SP,R0
022540  104417                                  TRAP    C$PNTF
022542  062706  000006                          ADD     #6,SP
022546                                          EXIT    DU
022546  000167                                  .WORD   J$JMP
022550  000030                                  .WORD   L10032-2-.
```

```
3995 022552     045     116     045 1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
3996                                 .EVEN
3997 022602                          ENDDU
     022602                     L10032:
     022602  104453                       TRAP    C$DU
3998                                 ;++
3999                                 ; AUTO-DROP CODE SECTION.
4000                                 ;--
4001 022604                          BGNAUTO
     022604                     L$AUTO::
4002 022604  012703  000550          MOV     #360.,R3        ;ENOUGH TIME FOR 2400' REEL TO REWIND
4003 022610  004737  016740     10$: JSR     PC,WAITF        ;WAIT FOR SSR TO SET
4004 022614  103420                 BCS     20$             ;LEAVE WHEN SSR IS SET
4005 022616                          DELAY   250.            ;WAIT FOR .25 SECONDS
     022616  012727  000372         MOV     #250.,(PC)+
     022622  000000                 .WORD   0
     022624  013727  002116         MOV     L$DLY,(PC)+
     022630  000000                 .WORD   0
     022632  005367  177772         DEC     -6(PC)
     022636  001375                 BNE     .-4
     022640  005367  177756         DEC     -22(PC)
     022644  001367                 BNE     .-20
4006 022646  005303                 DEC     R3              ;BUMP COUNTER DOWN
4007 022650  001357                 BNE     10$             ;KEEP GOING
4008 022652  004737  017772         JSR     PC,CKDROP       ;TRY AND DROP UNIT
4009 022656                     20$:
4010 022656                          ENDAUTO                 ; UNUSED.
     022656                     L10033:
     022656  104461                       TRAP    C$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.
                                        ;--
022660                                          BGNCLN
022660                                  L$CLEAN::
022660  005737  003060                          TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
022664  100405                                  BMI     1$              ;...AND GROSS CONTROLLER FAULT...
                                                                        ;...DON'T TRY TO XCT CLEANUP CODE.

022666  012765  000000  000000                  MOV     #0,TSSR(R5)     ;DO SOFT INIT
022674  004737  016740                          JSR     PC,WAITF
022700                                  1$:
022700                                  2$:     ENDCLN
022700                                  L10034:
022700  104412                                  TRAP    C$CLEAN

                                        ;++
                                        ; THE REPORT CODING SECTION CONTAINS THE
                                        ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                        ;--
022702                                          BGNRPT
022702                                  L$RPT::
022702                                          PRINTS  #DEVSUM
022702  012746  023144                          MOV     #DEVSUM,-(SP)
022706  012746  000001                          MOV     #1,-(SP)
022712  010600                                  MOV     SP,R0
022714  104416                                  TRAP    C$PNTS
022716  062706  000004                          ADD     #4,SP
022722  010246                                  MOV     R2,-(SP)
022724  010346                                  MOV     R3,-(SP)
022726  010446                                  MOV     R4,-(SP)
022730  012704  003130                          MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
022734  005003                                  CLR     R3              ; CLEAR UNIT NUMBER
022736  011402                          1$:     MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
022740  001467                                  BEQ     4$              ; ZERO IF UNIT NOT RUN
022742  100066                                  BPL     4$
022744  032702  040000                          BIT     #BIT14,R2       ; WAS UNIT DROPPED?
022750  001015                                  BNE     2$              ; BR IF YES
022752  042702  170000                          BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
022756                                          PRINTS  #DEVONL,R3,R2   ; PRINT
022756  010246                                  MOV     R2,-(SP)
022760  010346                                  MOV     R3,-(SP)
022762  012746  023201                          MOV     #DEVONL,-(SP)
022766  012746  000003                          MOV     #3,-(SP)
022772  010600                                  MOV     SP,R0
022774  104416                                  TRAP    C$PNTS
022776  062706  000010                          ADD     #10,SP
023002  000446                                  BR      4$
023004  020227  160000                  2$:     CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
023010  001012                                  BNE     3$              ; BR IF NO
023012                                          PRINTS  #DEVNXR,R3
023012  010346                                  MOV     R3,-(SP)
023014  012746  023251                          MOV     #DEVNXR,-(SP)
023020  012746  000002                          MOV     #2,-(SP)
```

```
                023024  010600                          MOV     SP,R0
                023026  104416                          TRAP    C$PNTS
                023030  062706  000006                  ADD     #6,SP
        4050    023034  000431                          BR      4$
        4051    023036  020227  160001          3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
        4052    023042  001012                          BNE     30$             ; BR IF NO.
        4053    023044                                  PRINTS  #DEVNRD,R3
                023044  010346                          MOV     R3,-(SP)
                023046  012746  023333                  MOV     #DEVNRD,-(SP)
                023052  012746  000002                  MOV     #2,-(SP)
                023056  010600                          MOV     SP,R0
                023060  104416                          TRAP    C$PNTS
                023062  062706  000006                  ADD     #6,SP
        4054    023066  000414                          BR      4$
        4055    023070  042702  170000          30$:    BIC     #^C7777,R2
        4056    023074                                  PRINTS  #DEVDRO,R3,R2
                023074  010246                          MOV     R2,-(SP)
                023076  010346                          MOV     R3,-(SP)
                023100  012746  023414                  MOV     #DEVDRO,-(SP)
                023104  012746  000003                  MOV     #3,-(SP)
                023110  010600                          MOV     SP,R0
                023112  104416                          TRAP    C$PNTS
                023114  062706  000010                  ADD     #10,SP
        4057    023120  062704  000002          4$:     ADD     #2,R4
        4058    023124  005203                          INC     R3
        4059    023126  020427  003330                  CMP     R4,#ERTABE
        4060    023132  103701                          BLO     1$
        4061    023134  012604                          MOV     (SP)+,R4
        4062    023136  012603                          MOV     (SP)+,R3
        4063    023140  012602                          MOV     (SP)+,R2
        4064    023142                                  ENDRPT                  ; UNUSED.
                023142                          L10035:
                023142  104425                          TRAP    C$RPT
        4065
        4066
        4067    023144      045     116     045 DEVSUM: .ASCIZ   /%N%ADEVICE STATUS SUMMARY:%N/
        4068    023201      045     101     040 DEVONL: .ASCIZ   /%A  UNIT %D3%A   ONLINE.  ERRORS = %D%N/
        4069    023251      045     101     040 DEVNXR: .ASCIZ   /%A  UNIT %D3%A   DROPPED, NON-EXISTENT REGISTER%N/
        4070    023333      045     101     040 DEVNRD: .ASCIZ   /%A  UNIT %D3%A   DROPPED, NOT READY AT STARTUP%N/
        4071    023414      045     101     040 DEVDRO: .ASCIZ   /%A  UNIT %D3%A   DROPPED, ERRORS = %D%N/
        4072                                            .EVEN
        4075
        4082
        4088
        4096
```

```
   4098                                                    .SBTTL  TEST  1: BUS RESET TEST
   4099
   4100
   4101                                            ;        THIS TEST VERIFIES THAT THE DV132 MODULE'S DEVICE REGISTERS ARE
   4102                                            ;        ACCESSIBLE ON THE BUS (SUBTEST 1) AND THEN CHECKS THAT THE
   4103                                            ;        BUILT-IN INITIALIZATION SELF-TEST MICRODIAGNOSTIC DID NOT FIND
   4104                                            ;        ANY BASIC PROBLEMS IN THE MODULE.  AREAS OF LOGIC TESTED BY THE
   4105                                            ;        SELF-TEST SEQUENCE ARE AS FOLLOWS:  ROM AND PIPELINE REGISTER,
   4106                                            ;        SEQUENCER, INTERNAL BUSES, 2901 MICROPROCESSOR, AND, RAM.  THIS
   4107                                            ;        TEST INITIALIZES THE CONTROLLER BY ISSUING THE BUS INIT SIGNAL
   4108                                            ;        VIA A RESET INSTRUCTION, OR BY WRITING INTO THE TSSR REGISTER,
   4109                                            ;        WAITS A PERIOD OF TIME (TO ALLOW THE CONTROLLER'S INITIALIZATION
   4110                                            ;        MICRODIAGNOSTIC SEQUENCE TO BE COMPLETED), AND THEN CHECKS THE
   4111                                            ;        CONTENTS OF THE TSSR REGISTER.  SUCCESSFUL INITIALIZATION IS
   4112                                            ;        INDICATED BY SUBSYSTEM READY (SSR) AND NEED BUFFER ADDRESS (NBA)
   4113                                            ;        BITS BEING SET (1) AND ALL OTHER BITS (EXCEPT A17 AND A16 AND
   4114                                            ;        OFL, WHICH ARE IGNORED FOR THIS TEST) BEING CLEAR (0).  IF THE
   4115                                            ;        CONTENTS OF TSSR ARE NOT AS EXPECTED, AN ERROR REPORT IS ISSUED
   4116                                            ;        LISTING THE EXPECTED DATA, ACTUAL DATA, AND THE DISCREPANCIES.
   4117                                            ;        THE ERROR REPORT ANALYZES THE TSSR CONTENTS AND DISCERNS AND
   4118                                            ;        REPORTS ONE OF THREE POSSIBILITIES:
   4119                                            ;
   4120                                            ;
   4121                                            ;             1.  TSSR CONTENTS ARE AMBIGUOUS (ANY OF BITS 11-14 ARE SET,
   4122                                            ;                 OR STATES OF SSR AND SC BITS DO NOT CORRESPOND TO THE
   4123                                            ;                 APPARENT ERROR CODE IN BITS 0-5):  INDICATES THAT THE
   4124                                            ;                 TSSR CONTENT CANNOT BE THUSTED.  INDICATES A
   4125                                            ;                 CATASTROPHIC CONTROLLER MALFUNCTION.  THIS IS A FATAL
   4126                                            ;                 ERROR (EXECUTION IS ABORTED).  FIELD ACTION WOULD BE TO
   4127                                            ;                 REPLACE THE DV132.  IF THE DV132 ITSELF IS BEING
   4128                                            ;                 DEBUGGED, THE PROGRAM SHOULD BE RESTARTED WITH LOOP ON
   4129                                            ;                 ERROR ENABLED IN ORDER TO PROBE FOR THE PROBLEM.
   4130                                            ;
   4131                                            ;             2.  SSR = 0, SC = 0 AND THE ERROR CODE IN BITS 0-5 IS IN
   4132                                            ;                 THE RANGE 17-13:  THIS IS A FATAL ERROR.  THE ERROR
   4133                                            ;                 CODE IS DECODED AND THE APPROPRIATE DESCRIPTION GIVEN.
   4134                                            ;                 INDICATES THAT A SERIOUS PROBLEM EXISTS.
   4135                                            ;
   4136                                            ;
   4137 023464                                             BGNTST
        023464                                                                                   T1::
   4138 023464  005037  002170                             CLR     FATFLG                   ;CLEAR FATAL ERROR FLAG
   4139 023470  012737  005672  002146                     MOV     #EPRT1,EPRTSW            ;SET UP ERROR MESSAGE SWITCH
   4140 023476  005037  003100                             CLR     KTFLG          ;HOLD OFF KT11
   4145 023502  012700  023700                             MOV     #TST1ID,R0               ;ASCII MESSAGE TO IDENTIFY TEST
   4146 023506  004737  017226                             JSR     PC,TSTSETUP              ;DO INITIAL TEST SETUP
   4147 023512  012737  000005  002164                     MOV     #5.,LOOPCNT              ;PERFORM 5 ITERATIONS
   4148 023520                                     T1LOOP:
   4149 023520  005003                                     CLR     R3                       ;USE R3 AS FATAL ERROR FLAG
   4150
   4151 023522                                             BGNSUB                           ;///////////// BEGIN SUBTEST ////////////
        023522                                                                                   T1.1:
        023522  104402                                                                                   TRAP    C$BSUB
   4152
   4153 023524                                             BRESET                           ;ISSUE A BUS RESET
        023524  104433                                                                                   TRAP    C$RESET
   4154 023526  004737  016740                             JSR     PC,WAITF                 ;WAIT FOR READY
```

```
4155 023532  016501  000000                  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
4156 023536  010102                          MOV     R1,R2                   ;CONTENTS OF TSSR
4157 023540  042702  176277                  BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
4158 023544  052702  002200                  BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
4159 023550  020102                          CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
4160 023552  001405                          BEQ     10$                     ;BRANCH IF COMPARE
4164 023554                                  ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
     023554  104455                                                                  TRAP    C$ERDF
     023556  000145                                                                  .WORD   101
     023560  003603                                                                  .WORD   SFHERR
     023562  011566                                                                  .WORD   SFFMSG
4165 023564  005203                          INC     R3                      ;SET THE FATAL ERROR FLAG
4166 023566                          10$:
4167 023566                                  ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     023566                                                                  L10037:
     023566  104403                                                                  TRAP    C$ESUB
4168
```

```
4170 023570  005703                    TST     R3              ;DID WE HAVE FATAL ERROR ?
4171 023572  001402                    BEQ     20$             ;BRANCH IF NOT
4172 023574  004737  017772            JSR     PC,CKDROP       ;GO DROP THIS UNIT, IF ALLOWED
4173 023600  005003              20$:  CLR     R3              ;RESET FATAL ERROR FLAG
4174
4175
4176 023602                            BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
     023602                                                                    T1.2:
     023602  104402                                                                    TRAP    C$BSUB
4177
4178 023604  005065  000000            CLR     TSSR(R5)        ;WRITE TO ISSUE A SOFT RESET
4179 023610  004737  016740            JSR     PC,WAITF        ;WAIT FOR READY TO SET
4180 023614  016501  000000            MOV     TSSR(R5),R1     ;GET REGISTER TSSR DATA
4181 023620  010102                    MOV     R1,R2           ;CONTENTS OF TSSR
4182 023622  042702  176277            BIC     #^C<HIADDR!OFL>,R2 ;THESE BITS MAY BE SET
4183 023626  052702  002200            BIS     #SSR!NBA,R2     ;READY AND NEW DATA SHOULD BE SET
4184 023632  020102                    CMP     R1,R2           ;COMPARE EXPECTED TO RECEIVED
4185 023634  001405                    BEQ     10$             ;BRANCH IF COMPARE
4189 023636                            ERRDF   ERRNO,SFIERR,SFFMSG ;REPORT A FATAL ERROR
     023636  104455                                                                    TRAP    C$ERDF
     023640  000146                                                                    .WORD   102
     023642  003550                                                                    .WORD   SFIERR
     023644  011566                                                                    .WORD   SFFMSG
4190 023646  005203                    INC     R3              ;SET THE ERROR FLAG
4191 023650                      10$:
4192 023650                            ENDSUB                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     023650                                                                    L10040:
     023650  104403                                                                    TRAP    C$ESUB
4193
4194
4195 023652  005703                    TST     R3              ;FATAL ERROR DETECTED ?
4196 023654  001402                    BEQ     20$             ;BRANCH IF NOT
4197 023656  004737  017772            JSR     PC,CKDROP       ;SEE IF TIME TO DROP UNIT
4198 023662  004737  017174      20$:  JSR     PC,TSTLOOP      ;SHOULD WE DO ITERATIONS ?
4199 023666  103002                    BCC     40$             ;BRANCH IF NOT
4200 023670  000137  023520            JMP     T1LOOP          ;LOOP UNTIL COUNT EXPIRED
4201 023674                      40$:  EXIT    TST             ;ALL DONE THIS TEST
     023674  104432                                                                    TRAP    C$EXIT
     023676  000022                                                                    .WORD   L10036-.
4202
4203                                   ;+
4204                                   ;LOCAL TEXT MESSAGES FOR TEST
4205                                   ;-
4206
4207 023700     111     156     151   TST11D: .ASCIZ  'Initialization'
4208                                           .EVEN
4209 023720                                    ENDTST
     023720                                                                  L10036:
     023720  104401                                                                  TRAP    C$ETST
4210
4211
```

```
                                .SBTTL  TEST  2: RAM TEST

                        ;       THIS TEST VERIFIES THAT ALL LOCATIONS OF THE RAM ON THE DV132
                        ;       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
                        ;       DV132 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.

023722                  BGNTST                                  T2::
023722
023722  005037  002170            CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
023726  012737  005672  002146    MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
023734  005037  003100            CLR     KTFLG           ;HOLD OFF KT11

023740                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
023740                                                          T2.1:
023740  104402                                                          TRAP    C$BSUB
```

TEST 2: RAM TEST

```
4272 023742  012700  024666                    MOV     #TST2ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
4273 023746  004737  017226                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
4274 023752  012737  000002  002164            MOV     #2,LOOPCNT              ;PERFORM 2 ITERATIONS
4275 023760                          T2LOOP:
4276 023760  004737  016464                    JSR     PC,SDFINIT              ;DO INITIALIZE ON CONTROLLER
4277 023764  103405                            BCS     20$                     ;BR IF INIT WAS OK
4281 023766  010001                            MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
4282 023770                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
     023770  104455                                                                    TRAP    C$ERDF
     023772  000311                                                                    .WORD   201
     023774  003550                                                                    .WORD   SFIERR
     023776  011506                                                                    .WORD   SFIMSG
4283 024000  012704  000002            20$:    MOV     #2,R4                   ;SET RAM ADDRESS AT TWO
4284 024004                             25$:
4285
4286 024004  110402                            MOVB    R4,R2                   ;EXPECTED DATA FROM WRAP-AROUND
4287 024006  110465  177777                    MOVB    R4,TSDBH(R5)            ;LOAD ADDRESS INTO TSDB
4288 024012  110265  177776                    MOVB    R2,TSDBL(R5)            ;LOADS DATA INTO RAM LOCATION
4289 024016  116501  177776                    MOVB    TSBAL(R5),R1            ;READS WRAP DATA
4290 024022  120102                            CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ
4291 024024  001404                            BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
4295 024026                                    ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
     024026  104456                                                                    TRAP    C$ERHRD
     024030  000312                                                                    .WORD   202
     024032  024524                                                                    .WORD   TSBAM2
     024034  016164                                                                    .WORD   EXPREC
4296 024036                             30$:
4297
4298 024036  005204                            INC     R4                      ;NEXT ADDRESS
4299 024040  020427  000400                    CMP     R4,#400                 ;END OF RAM MEMORY CHECK
4300 024044  001357                            BNE     25$                     ;LOOP TILL ALL RAM WRITTEN
4301 024046  005002                            CLR     R2                      ;CLEAR OUT R2 HIGH BITS
4302 024050  005304                            DEC     R4                      ;SET BACK TO 377
4303 024052  110402                    40$:    MOVB    R4,R2                   ;GET DATA PATTERN BACK IN SHAPE
4304 024054  110465  177777                    MOVB    R4,TSDBH(R5)            ;LOAD UP RAM ADDRESS POINTER
4305 024060  116501  177776                    MOVB    TSBAL(R5),R1            ;READ RAM CONTENTS BACK
4306 024064  120102                            CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
4307 024066  001404                            BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
4311 024070                                    ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
     024070  104456                                                                    TRAP    C$ERHRD
     024072  000313                                                                    .WORD   203
     024074  024524                                                                    .WORD   TSBAM2
     024076  016164                                                                    .WORD   EXPREC
4312 024100                             45$:    CKLOOP                          ;SCOPE LOOP
     024100  104406                                                                    TRAP    C$CLP1
4313 024102  005304                            DEC     R4                      ;DROP DATA COUNTER (PATTERN)
4314 024104  022704  000002                    CMP     #2,R4                   ;AT LOC TWO YET
4315 024110  001360                            BNE     40$                     ;BR, IF NOT AT TWO YET
4316
4317 024112                                    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     024112                                                                                L10042:
     024112  104403                                                                    TRAP    C$ESUB
4318
```

```
     024114                             BGNSUB                          ;///////////// BEGIN SUBTEST ////////////
     024114                                                                            T2.2:
     024114  104402                                                                                  TRAP    C$BSUB
                                  ;     TEST 2, SUBiEST 2

                                  ;
                                  ;     THIS SUBTEST WRITES RAM WITH ALL ZEROS
                                  ;     THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
                                  ;
     024116  004737  016464             JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
     024122  103405                     BCS     20$                     ;BR IF INIT WAS OK
     024124  010001                     MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
     024126                             ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
     024126  104455                                                                          TRAP    C$ERDF
     024130  000314                                                                          .WORD   204
     024132  003550                                                                          .WORD   SFIERR
     024134  011506                                                                          .WORD   SFIMSG
     024136  005002               20$:  CLR     R2                      ;iEST DATA = 0
     024140  012704  000002             MOV     #2,R4                   ;STARTING RAM ADDRESS = 2
     024144                       25$:

     024144  110465  177777             MOVB    R4,TSDBH(R5)            ;LOAD ADDRESS INTO TSDB
     024150  110265  177776             MOVB    R2,TSDBL(R5)            ;LOADS DATA INTO RAM LOCATION
     024154  116501  177776             MOVB    TSBAL(R5),R1            ;READS WRAP DATA
     024160  120102                     CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
     024162  001404                     BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
     024164                             ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
     024164  104456                                                                          TRAP    C$ERHRD
     024166  000315                                                                          .WORD   205
     024170  024524                                                                          .WORD   TSBAM2
     024172  016164                                                                          .WORD   EXPREC
     024174                       30$:

     024174  005204                     INC     R4                      ;NEXT ADDRESS
     024176  020427  000400             CMP     R4,#400                 ;END OF RAM MEMORY CHECK
     024202  001360                     BNE     25$                     ;BR, MORE RAM TO GO

     024204  005304               35$:  DEC     R4                      ;SET BACK TO 377
     024206  005002                     CLR     R2                      ;SET TO ALL ZEROS
     024210                       40$:
     024210  110465  177777             MOVB    R4,TSDBH(R5)            ;LOAD UP THE ADDRESS FOR RAM
     024214  116501  177776             MOVB    TSBAL(R5),R1            ;READ THE RAM CONTENTS BACK
     024220  005002                     CLR     R2                      ;LOOKING FOR 000000 (EXPECTED)
     024222  120102                     CMPB    R1,R2                   ;BOTH SHOULD BE 00000000 BINARY
     024224  001404                     BEQ     43$                     ;BR, IF DATA IS GOOD
     024226                             ERRHRD  ERRNO,TSBAM3,EXPREC     ;CHARACTERISTICS DATA NOT CORRECT
     024226  104456                                                                          TRAP    C$ERHRD
     024230  000316                                                                          .WORD   206
     024232  024606                                                                          .WORD   TSBAM3
     024234  016164                                                                          .WORD   EXPREC
     024236  012702  000377       43$:  MOV     #000377,R2              ;SET ALL ONES WORD
     024242  110465  177777             MOVB    R4,TSDBH(R5)            ;LOAD UP RAM ADDRESS POINTER
     024246  110265  177776             MOVB    R2,TSDBL(R5)            ;WRITE DATA INTO RAM
     024252  116501  177776             MOVB    TSBAL(R5),R1            ;READ RAM CONTENTS BACK
     024256  120102                     CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
     024260  001404                     BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
```

```
4375 024262                              ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
     024262  104456                                                              TRAP    C$ERHRD
     024264  000317                                                              .WORD   207
     024266  024524                                                              .WORD   TSBAM2
     024270  016164                                                              .WORD   EXPREC
4376 024272                         45$: CKLOOP                                 ;SCOPE LOOP
     024272  104406                                                              TRAP    C$CLP1
4377 024274  005304                      DEC     R4                             ;DROP RAM ADDRESS POINTER
4378 024276  022704  000002              CMP     #2,R4                          ;AT LOC 2 YET
4379 024302  001342                      BNE     40$                            ;BR, IF NOT AT TWO YET
4380
4381 024304                              ENDSUB                                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     024304                                                                  L10043:
     024304  104403                                                              TRAP    C$ESUB
4382
```

```
024306                          BGNSUB                              ;///////////// BEGIN SUBTEST ////////////
024306                                                                  T2.3:
024306  104402                                                                      TRAP    C$BSUB

                        ;       TEST 2, SUBTEST 3
                        ;
                        ;       THIS SUBTEST WRITES RAM WITH ALL ONES
                        ;       THEN WALKS A ZERO WORD DOWN THROUGH MEMORY
                        ;
024310  004737  016464          JSR     PC,SOFINIT                  ;DO INITIALIZE ON CONTROLLER
024314  103405                  BCS     20$                         ;BR IF INIT WAS OK
024316  010001                  MOV     R0,R1                       ;CONTENTS OF TSSR REGISTER
024320                          ERRDF   ERRNO,SFIEPR,SFIMSG         ;FATAL ERROR TSSR WAS NOT OK
024320  104455                                                                      TRAP    C$ERDF
024322  000320                                                                      .WORD   208
024324  003550                                                                      .WORD   SFIERR
024326  011506                                                                      .WORD   SFIMSG
024330  012702  177777  20$:    MOV     #177777,R2                  ;SET DATA AT ALL ONES
024334  012704  000002          MOV     #2,R4                       ;SET RAM ADDRESS AT TWO
024340                  25$:
024340                          BGNSEG                              ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
024340  104404                                                                      TRAP    C$BSEG
024342  110465  177777          MOVB    R4,TSDBH(R5)                ;LOAD ADDRESS INTO TSDB
024346  110265  177776          MOVB    R2,TSDBL(R5)                ;LOADS DATA INTO RAM LOCATION
024352  116501  177776          MOVB    TSBAL(R5),R1                ;READS WRAP DATA
024356  120102                  CMPB    R1,R2                       ;DOES WRITTEN(WRAP) = READ ?
024360  001404                  BEQ     30$                         ;BR IF OK, THEY ARE EQUAL
024362                          ERRHRD  ERRNO,TSBAM2,EXPREC         ;DATA NOT WRAPPED CORRECTLY
024362  104456                                                                      TRAP    C$ERHRD
024364  000321                                                                      .WORD   209
024366  024524                                                                      .WORD   TSBAM2
024370  016164                                                                      .WORD   EXPREC
024372                  30$:
024372                          ENDSEG                              ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
024372                                                                  10000$:
024372  104405                                                                      TRAP    C$ESEG

024374  005204                  INC     R4                          ;NEXT ADDRESS
024376  020427  000400          CMP     R4,#400                     ;END OF RAM MEMORY CHECK
024402  001356                  BNE     25$                         ;BR, MORE RAM TO GO

024404  005304          35$:    DEC     R4                          ;SET BACK TO 377
024406                  40$:
024406  012702  000377          MOV     #000377,R2                  ;SET UP EXPECTED DATA REGISTER
024412  005001                  CLR     R1                          ;CLEAN OUT REGISTER
024414  110465  177777          MOVB    R4,TSDBH(R5)                ;SELECT ADDRESS IN RAM
024420  116501  177776          MOVB    TSBAL(R5),R1                ;PICK UP RAM CONTENTS
024424  120102                  CMPB    R1,R2                       ;IS MEMORY STILL ALL ONES
024426  001404                  BEQ     43$                         ;BR, IF OK (ALL ONES)
024430                          ERRHRD  ERRNO,TSBAM3,EXPREC         ;MEMORY CHANGED AFTER ALL ONES WRITE
024430  104456                                                                      TRAP    C$ERHRD
024432  000322                                                                      .WORD   210
024434  024606                                                                      .WORD   TSBAM3
024436  016164                                                                      .WORD   EXPREC
024440  005002          43$:    CLR     R2                          ;SET UP NEW EXPECTED
024442  110465  177777          MOVB    R4,TSDBH(R5)                ;LOAD UP RAM ADDRESS POINTER
```

```
4433 024446  110265  177776                MOVB    R2,TSDBL(R5)          ;WRITE DATA INTO RAM
4434 024452  116501  177776                MOVB    TSBAL(R5),R1          ;READ RAM CONTENTS BACK
4435 024456  120102                        CMPB    R1,R2                 ;CHECK WITH DATA WRITTEN
4436 024460  001404                        BEQ     45$                   ;BR IF OK, DATA IN = DATA OUT
4440 024462                                ERRHRD  ERRNU,TSBAM2,EXPREC   ;WRITTEN DATA NOT = TO READ
     024462  104456                                                              TRAP    C$ERHRD
     024464  000323                                                              .WORD   211
     024466  024524                                                              .WORD   TSBAM2
     024470  016164                                                              .WORD   EXPREC
4441 024472                         45$:    CKLOOP                        ;SCOPE LOOP
     024472  104406                                                              TRAP    C$CLP1
4442 024474  005304                        DEC     R4                    ;DROP RAM ADDRESS POINTER
4443 024476  022704  000002                CMP     #2,R4                 ;CHECK LOC TWO
4444 024502  001341                        BNE     40$                   ;BR, IF NOT AT LOC 2 YET
4445
4446 024504                                ENDSUB                        ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     024504                                                                   L10044:
     024504  104403                                                              TRAP    C$ESUB
4447
4448 024506  004737  017174                JSR     PC,TSTLOOP            ;DO WE NEED TO ITERATE TEST ?
4449 024512  103002                        BCC     63$                   ;BRANCH IF NOT
4450 024514  000137  023760                JMP     T2LOOP                ;EXECUTE AGAIN
4451 024520                         63$:    EXIT    TST                   ;ALL DONE THIS TEST
     024520  104432                                                              TRAP    C$EXIT
     024522  000150                                                              .WORD   L10041-.
4452                                ;+
4453                                ;LOCAL TEXT MESSAGES FOR TEST
4454                                ;-
4455
4456 024524     040     127     162  TSBAM2: .ASCIZ  ' Write to TSDB Not Equal to Read of TSBA Low Byte'
4457 024606     127     162     151  TSBAM3: .ASCIZ  'Write To RAM Location Modified Another Location'
4458 024666     122     141     155  TST2ID: .ASCIZ  'Ram'
4459                                        .EVEN
4460 024672                                ENDTST
     024672                                                                   L10041:
     024672  104401                                                              TRAP    C$ETST
```

```
4462                                    .SBTTL  TEST  3: COMMAND REJECT
4463
4464
4465                                    ;       THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE
4466                                    ;       CHARACTERISTICS ARE REJECTED DUE TO THE NEED BUFFER ADDRESS
4467                                    ;       (NBA) BIT BEING SET IN TSSR, AND THAT THE TSBA AND TSSR
4468                                    ;       REGISTERS ARE LEFT IN THE PROPER STATE AFTER EACH COMMAND IS
4469                                    ;       REJECTED.  THIS TEST CHECKS MICROPROCESSOR SEQUENCING, BASIC
4470                                    ;       COMMAND DECODING AND DATI DMA HANDLING.  THIS TEST CONTAINS TWO
4471                                    ;       SUBTESTS:  SUBTEST 1 SEQUENCES THROUGH ALL COMMAND WORDS (OTHER
4472                                    ;       THAN WRITE CHARACTERISTICS) WITH THE INTERRUPT ENABLE (IE) BIT
4473                                    ;       CLEAR AND VERIFIES THAT AN INTERRUPT IS NOT GENERATED BY THE
4474                                    ;       REJECTED COMMAND;  SUBTEST 2 PERFORMS SIMILARLY TO SUBTEST 1 BUT
4475                                    ;       SETS THE IE BIT IN EACH COMMAND WORD AND VERIFIES THAT AN
4476                                    ;       INTERRUPT IS GENERATED WHEN THE COMMAND IS REJECTED.  SUBTEST 1
4477                                    ;       SETS UP THE INTERRUPT SERVICE ROUTINE TO FLAG UNEXPECTED
4478                                    ;       INTERRUPTS.  THE COMMAND WORD IN THE COMMAND BUFFER IS
4479                                    ;       INITIALIZED TO 100000 (OCTAL) AND THE REMAINING THREE WORDS IN
4480                                    ;       THE COMMAND BUFFER ARE SET TO KNOWN UNIQUE PATTERNS.  THEN THE
4481                                    ;       FOLLOWING SEQUENCE IS PERFORMED:
4482                                    ;
4483                                    ;            1.  INITIALIZE THE CONTROLLER BY WRITING INTO THE TSSR;
4484                                    ;                PROPER INITIAL CONDITIONS ARE VERIFIED.
4485                                    ;
4486                                    ;            2.  TSDB IS WRITTEN WITH ADDRESS OF THE COMMAND BUFFER TO
4487                                    ;                START PROCESSING.
4488                                    ;
4489                                    ;            3.  THE PROGRAM WAITS FOR SSR TO SET;  IF SSR DOES NOT SET,
4490                                    ;                AN ERROR REPORT IS ISSUED AND THE TEST IS ABORTED.
4491                                    ;
4492                                    ;            4.  THE CONTENTS OF TSSR ARE CHECKED.  TSSR IS CORRECT IF
4493                                    ;                IT CONTAINS EITHER OCTAL 102206 OR 102306 (BIT 6
4494                                    ;                DEPENDS UPON THE STATE OF THE TAPE TRANSPORT).
4495                                    ;
4496                                    ;            5.  THE CONTENTS OF TSBA ARE CHECKED.  TSBA SHOULD CONTAIN
4497                                    ;                THE INITIAL COMMAND BUFFER ADDRESS (LOADED IN STEP 2)
4498                                    ;                PLUS 10 (OCTAL);  I.E., TSBA SHOULD POINT TO THE WORD
4499                                    ;                JUST AFTER THE COMMAND PACKET (NOTE THAT 4 COMMAND
4500                                    ;                PACKET WORDS ARE ALWAYS FETCHED).
4501                                    ;
4502                                    ;            6.  USING THE MAINTENANCE MODE WRAPAROUND FUNCTIONS, THE
4503                                    ;                COMMAND IMAGE BLOCK IN THE DV132'S RAM (LOCATIONS
4504                                    ;                201-210 (OCTAL)) ARE CHECKED;  THE IMAGE SHOULD CONTAIN
4505                                    ;                A COPY OF THE FOUR COMMAND PACKET WORDS AS SET UP IN
4506                                    ;                CPU MEMORY.
4507                                    ;
4508                                    ;            7.  THE COMMAND WORD IN THE COMMAND BUFFER IS INCREMENTED
4509                                    ;                TO THE NEXT PATTERN NOT CONTAINING WRITE
4510                                    ;                CHARACTERISTICS OR IE.  THE REMAINING THREE WORD OF THE
4511                                    ;                COMMAND BUFFER ARE SEQUENCED WITH PSEUDO-RANDOM DATA.
4512                                    ;                IF THE COMMAND WORD HAS NOT REACHED ITS MAXIMUM VALUE
4513                                    ;                (177777+1), THE TEST SEQUENCE IS REPEATED.
4514
4515
4516
4517
4518                                    ;       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
                        T3LOOP:
                                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 TSSR
                                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   I3INI
```

```
                025066  011520                                                                  .WORD   PKTSSR
4569            025070  012702  102206          22$:    MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
4570            025074  004737  017054                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
4571            025100  016501  000000                  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
4572            025104  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
4573            025110  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
4574            025112  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
4575            025116  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
4576            025120  001404                          BEQ     30$                     ;DOES EXPECTED MATCH RECEIVED?
4580            025122                                  ERRHRD  ERRNO,T3NBA,PKTSSR      ;NBA NOT SET TO REJECT
                025122  104456                                                                  TRAP    C$ERHRD
                025124  000460                                                                  .WORD   304
                025126  025620                                                                  .WORD   T3NBA
                025130  011520                                                                  .WORD   PKTSSR
4581            025132                          30$:    CKLOOP                          ;LOOP ON ERROR ?
                025132  104406                                                                  TRAP    C$CLP1
4582            025134  004737  017054                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
4583            025140  016501  177776                  MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
4584            025144  010402                          MOV     R4,R2                   ;START OF THE PACKET
4585            025146  020102                          CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
4586            025150  001404                          BEQ     35$                     ;ERROR IF NOT EQUAL
4590            025152                                  ERRHRD  ERRNO,T3TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
                025152  104456                                                                  TRAP    C$ERHRD
                025154  000461                                                                  .WORD   305
                025156  026061                                                                  .WORD   T3TSBA
                025160  016164                                                                  .WORD   EXPREC
4591
4592
4593            025162  004737  010354          35$:    JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
4594            025166  103404                          BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
4598            025170                                  ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
                025170  104456                                                                  TRAP    C$ERHRD
                025172  000462                                                                  .WORD   306
                025174  004643                                                                  .WORD   PKTRAM
                025176  016200                                                                  .WORD   RAMERR
4599            025200                          40$:    ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                025200                                                                  10000$:
                025200  104405                                                                  TRAP    C$ESEG
4600            025202  011300                          MOV     (R3),R0                 ;PACKET COMMAND WORD
4601            025204  042700  177740                  BIC     #177740,R0              ;GET BITS 0-4
4602            025210  020027  000004                  CMP     R0,#4                   ;DON'T TEST WRITE CHARACTERISTICS
4603            025214  001002                          BNE     45$                     ;BRANCH IF OK
4604            025216  062703  000002                  ADD     #2,R3                   ;GET NEXT WORD FROM DATA TABLE
4605            025222  020327  003030          45$:    CMP     R3,#TBLEND              ;REACHED END OF TABLE ?
4606            025226  103002                          BHIS    50$                     ;BRANCH IF END OF TABLE
4607            025230  000137  024750                  JMP     5$                      ;CONTINUE TEST WITH NEW DATA
4608
4609            025234                          50$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                025234                                                                  L10046:
                025234  104403                                                                  TRAP    C$ESUB
4610
4611            025236  005737  002170                  TST     FATFLG                  ;ANY FATAL ERRORS ?
4612            025242  001402                          BEQ     60$                     ;BRANCH IF NOT
4613            025244  004737  017772                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
```

```
4615 025250                          60$:   BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     025250                                                                                    T3.2:
     025250  104402                                                                                 TRAP    C$BSUB
4616
4617 025252                                 SETPRI  #PR100                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     025252  012700  000000                                                                         MOV     #PR100,R0
     025256  104441                                                                                 TRAP    C$SPRI
4618 025260  012704  025610                 MOV     #T3PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
4619 025264  012703  002720                 MOV     #TSTBLK,R3              ;START OF TEST DATA
4620 025270  012314                 5$:     MOV     (R3)+,(R4)              ;PLACE NEXT DATA WORD IN PACKET
4621 025272                                 BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     025272  104404                                                                                 TRAP    C$BSEG
4622 025274  004737  016464                 JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
4623 025300  103405                         BCS     10$                     ;BR IF SOFT INIT = OK
4627 025302  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4628 025304                                 ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     025304  104455                                                                                 TRAP    C$ERDF
     025306  000463                                                                                 .WORD   307
     025310  003550                                                                                 .WORD   SFIERR
     025312  011506                                                                                 .WORD   SFIMSG
4629 025314  005037  002170         10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
4630 025320  005037  002172                 CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
4631 025324  004737  017054                 JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
4632 025330  052714  000200                 BIS     #BIT7,(R4)              ;ENABLE INTERRUPTS
4633 025334  010465  177776                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
4634 025340  004737  016740                 JSR     PC,WAITF                ;WAIT FOR SSR TO SET
4635 025344  103407                         BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
4636 025346  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4640 025350                                 ERRDF   ERRNO,T3SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     025350  104455                                                                                 TRAP    C$ERDF
     025352  000464                                                                                 .WORD   308
     025354  025645                                                                                 .WORD   T3SSR
     025356  011520                                                                                 .WORD   PKTSSR
4641 025360  004737  017720                 JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4642 025364                         15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     025364  104406                                                                                 TRAP    C$CLP1
4643 025366                                 ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     025366  104410                                                                                 TRAP    C$ESCAPE
     025370  000164                                                                                 .WORD   L10047-.
4644 025372  005737  002172                 TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
4645 025376  001004                         BNE     22$                     ;BRANCH IF YES
4649 025400                                 ERRHRD  ERRNO,T3NINT,PKTSSR     ;REPORT ERROR IF NO INTERRUPT
     025400  104456                                                                                 TRAP    C$ERHRD
     025402  000465                                                                                 .WORD   309
     025404  026001                                                                                 .WORD   T3NINT
     025406  011520                                                                                 .WORD   PKTSSR
4650 025410  012702  102206         22$:    MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
4651 025414  004737  017054                 JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
4652 025420  016501  000000                 MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
4653 025424  032701  000100                 BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
4654 025430  001402                         BEQ     25$                     ;BRANCH IF NOT OFF-LINE
4655 025432  052702  000100                 BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
4656 025436  020201                 25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
4657 025440  001404                         BEQ     30$                     ;OKAY IF MATCH
4661 025442                                 ERRHRD  ERRNO,T3NBA,PKTSSR      ;NBA NOT SET TO REJECT
     025442  104456                                                                                 TRAP    C$ERHRD
     025444  000466                                                                                 .WORD   310
```

```
                                                                                .WORD   T3NBA
                                                                                .WORD   PKTSSR
4662 025452                             30$:    CKLOOP                          ;LOOP ON ERROR ?
     025452 104406                                                              TRAP    C$CLP1
4663 025454 004737 017054                       JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
4664 025460 016501 177776                       MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
4665 025464 010402                              MOV     R4,R2                   ;START OF THE PACKET
4666 025466 020102                              CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
4667 025470 001404                              BEQ     35$                     ;ERROR IF NOT EQUAL
4671 025472                                     ERRHRD  ERRNO,T3TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     025472 104456                                                              TRAP    C$ERHRD
     025474 000467                                                              .WORD   311
     025476 026061                                                              .WORD   T3TSBA
     025500 016164                                                              .WORD   EXPREC
4672
4673
4674 025502 004737 010354               35$:    JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
4675 025506 103404                              BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
4679 025510                                     ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     025510 104456                                                              TRAP    C$ERHRD
     025512 000470                                                              .WORD   312
     025514 004643                                                              .WORD   PKTRAM
     025516 016200                                                              .WORD   RAMERR
4680 025520                             40$:    ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     025520                                                                     10000$:
     025520 104405                                                              TRAP    C$ESEG
4681 025522 011300                              MOV     (R3),R0                 ;NEXT PACKET COMMAND WORD
4682 025524 042700 177740                       BIC     #177740,R0              ;GET BITS 0-4
4683 025530 020027 000004                       CMP     R0,#4                   ;DON'T TEST WRITE CHARACTERISTICS
4684 025534 001002                              BNE     45$                     ;BRANCH IF NOT WRITE CHARACTERISTICS
4685 025536 062703 000002                       ADD     #2,R3                   ;BY-PASS WRITE CHARACTERISTCS
4686 025542 020327 003030               45$:    CMP     R3,#TBLEND              ;HAVE WE COMPLETED DATA TABLE ?
4687 025546 103002                              BHIS    50$                     ;BRANCH IF ALL TESTED
4688 025550 000137 025270                       JMP     5$                      ;TEST WITH NEXT DATA
4689
4690 025554                             50$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     025554                                                                     L10047:
     025554 104403                                                              TRAP    C$ESUB
4691 025556 005737 002170                       TST     FATFLG                  ;ANY FATAL ERRORS ?
4692 025562 001402                              BEQ     60$                     ;BRANCH IF NOT
4693 025564 004737 017772                       JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4694 025570 004737 017174               60$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
4695 025574 103002                              BCC     62$                     ;BRANCH IF NOT
4696 025576 000137 024730                       JMP     T3LOOP                  ;LOOP UNTIL COUNT EXPIRED
4697 025602                             62$:    EXIT    TST                     ;ALL DONE THIS TEST
     025602 104432                                                              TRAP    C$EXIT
     025604 000346                                                              .WORD   L10045-.
4698
4699                                    ;+
4700                                    ;LOCAL STORAGE FOR THIS TEST
4701                                    ;-
4702
4704 025606                                     .BLKB   10-<.-TUV2A&7>
4706 025610                             T3PACKET:                               ;COMMAND PACKET FOR TEST
4707 025610 000000                              .WORD   0                       ;WILL CONTAIN VARIABLE COMMANDS
4708 025612 052525                              .WORD   052525
4709 025614 125252                              .WORD   125252
```

```
025616  052525                              .WORD   052525


                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

025620     103      157      155  T3NBA:   .ASCIZ  'Command Not Rejected'
025645     103      157      156  T3SSR:   .ASCIZ  'Contents of TSSR Incorrect After Write Packet'
025723     125      156      145  T3INT:   .ASCIZ  'Unexpected Interrupt Received On Write Packet'
026001     105      170      160  T3NINT:  .ASCIZ  'Expected Interrupt Not Received On Write Packet'
026061     111      156      143  T3TSBA:  .ASCIZ  'Incorrect TSBA Address After Packet Write'
026133     103      157      155  TST3ID:  .ASCIZ  'Command Reject'
                                           .EVEN
026152                                     ENDTST
026152                                                                    L10045:
026152  104401                                                                    TRAP    C$ETST
```

```
4726
4727                                    .SBTTL  TEST  4: WRITE CHARACTERISTICS
4728
4729
4730                            ;       THIS TEST VERIFIES BASIC OPERATION OF THE WRITE CHARACTERISTICS
4731                            ;       COMMAND.  IT VERIFIES THAT THE COMMAND BLOCK AND CHARACTERISTICS
4732                            ;       DATA BLOCK ARE FETCHED PROPERLY FROM CPU MEMORY, THE NEED BUFFER
4733                            ;       ADDRESS (NBA) BIT IN TSSR IS HANDLED PROPERLY, AND THAT A PROPER
4734                            ;       MESSAGE PACKET IS STORED, WHERE APPROPRIATE.  THIS TEST DOES NOT
4735                            ;       CHECK THAT THE VARIOUS FUNCTIONS ENABLED BY CHARACTERISTIC MODE
4736                            ;       DATA BITS OPERATE PROPERLY;  THE FUNCTIONING OF THESE BITS IS
4737                            ;       VERIFIED IN SUBSEQUENT TESTS.  ALL COMMANDS EXECUTED IN THIS
4738                            ;       TEST HAVE THE INTERRUPT ENABLE (IE) BIT CLEARED TO ZERO, SO NO
4739                            ;       INTERRUPTS SHOULD BE GENERATED.  HOWEVER, THE PROGRAM RUNS AT
4740                            ;       PROCESSOR PRIORITY 0, WITH THE INTERRUPT SERVICE ROUTINE SET UP
4741                            ;       TO FLAG UNEXPECTED INTERRUPTS.  IF AN INTERRUPT OCCURS, A
4742                            ;       PROBLEM EXISTS IN EITHER THE LSI-11 BUS INTERFACE SECTION OR IN
4743                            ;       THE ROM OR PIPELINE.
4744                            ;
4745                            ;       THIS TEST CHECKS VARIOUS MICROPROGRAM SEQUENCES, COMMAND
4746                            ;       DECODING, DMA LOGIC, AND BASIC PACKET PROTOCOL HANDLING.  THIS
4747                            ;       IS THE FIRST TEST IN WHICH DATO DMA CYCLES (FOR STORING THE
4748                            ;       MESSAGE PACKET) ARE PERFORMED.  ANY ERRORS IN THE BODY OF THE
4749                            ;       TEST (I.E. ERRORS OTHER THAN INITIALIZATION ERRORS RELATED TO
4750                            ;       THE TRANSPORT BUS) DEFINITELY INDICATE A BAD M7454 MODULE.
4751                            ;
4752                            ;
4753                            ;
4754 026154                            BGNTST
      026154                                                                  T4::
4755 026154  005037  002170            CLR     @ATFLG                  ;CLEAR FATAL ERROR FLAG
4756 026160  012737  005672  002146    MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
4757 026166  005037  003100            CLR     KTFLG           ;HOLD OFF KT11
4762 026172  012700  030107            MOV     #TST4ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
4763 026176  004737  017226            JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
4764 026202  012737  000002  002164    MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
4765 026210                    T4LOOP:
4766 026210                            BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
      026210                                                                  T4.1:
      026210  104402                                                           TRAP    C$BSUB
4767 026212  004737  030136            JSR     PC,T4REST               ;SET PACKET TO START-UP VALUES
4768
4769 026216                            SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      026216  012700  000000                                                    MOV     #PRI00,R0
      026222  104441                                                            TRAP    C$SPRI
4770 026224  012703  002732            MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
4771 026230  012704  027240            MOV     #T4PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
4772 026234  012764  000010  000006    MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
4773 026242                    5$:
4774 026242                            BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      026242  104404                                                            TRAP    C$BSEG
4775
4776 026244  004737  016464            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
4777 026250  103405                    BCS     10$                     ;BR IF SOFT INIT = OK
4781 026252  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4782 026254                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      026254  104455                                                            TRAP    C$ERDF
```

```
                026256  000621                                                          .WORD   401
                026260  003550                                                          .WORD   SFIERR
                026262  011506                                                          .WORD   SFIMSG
  4783 026264  005037  002170          10$:    CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
  4784 026270  005037  002172                  CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
  4785 026274  010465  177776                  MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
  4786 026300  004737  017054                  JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
  4787 026304  103407                          BCS     15$             ;BR IF CARRY SET (GOOD RETURN)
  4788 026306  010001                          MOV     R0,R1           ;SAVE CONTENTS OF TSSR
  4792 026310                                  ERRDF   ERRNO,T4SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                026310  104455                                                          TRAP    C$ERDF
                026312  000622                                                          .WORD   402
                026314  027646                                                          .WORD   T4SSR
                026316  011520                                                          .WORD   PKTSSR
  4793 026320  004737  017720                  JSR     PC,FATCHK       ;INC AND CHECK FOR MORE THAN 25 ERRORS
  4794 026324          104406          15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                                                                                        TRAP    C$CLP1
  4795 026326                                  ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
                026326  104410                                                          TRAP    C$ESCAPE
                026330  000126                                                          .WORD   10000$-.
  4796 026332  005737  002172                  TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
  4797 026336  001404                          BEQ     22$             ;BRANCH IF NOT
  4801 026340                                  ERRHRD  ERRNO,T4INT,PKTSSR
                026340  104456                                                          TRAP    C$ERHRD
                026342  000623                                                          .WORD   403
                026344  027735                                                          .WORD   T4INT
                026346  011520                                                          .WORD   PKTSSR
  4802 026350  016501  000000          22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
  4803 026354  012702  000200                  MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
  4804 026360  032701  000100                  BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
  4805 026364  001402                          BEQ     25$             ;BRANCH IF NOT OFF-LINE
  4806 026366  052702  000100                  BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
  4807 026372  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
  4808 026374  001404                          BEQ     30$             ;OKAY IF MATCH
  4812 026376                                  ERRHRD  ERRNO,T4NBA,PKTSSR      ;NBA NOT ZERO
                026376  104456                                                          TRAP    C$ERHRD
                026400  000624                                                          .WORD   404
                026402  027376                                                          .WORD   T4NBA
                026404  011520                                                          .WORD   PKTSSR
  4813 026406          104406          30$:    CKLOOP                  ;LOOP ON ERROR ?
                                                                                        TRAP    C$CLP1
  4814 026410  004737  017054                  JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
  4815 026414  016501  177776                  MOV     TSBA(R5),R1     ;GET TSBA REGISTER CONTENTS
  4816 026420  012702  027240                  MOV     #T4PACKET,R2    ;START OF THE BUFFER
  4817 026424  020102                          CMP     R1,R2           ;COMPARE EXPECTED TO RECEIVED
  4818 026426  001404                          BEQ     35$             ;ERROR IF NOT EQUAL
  4822 026430                                  ERRHRD  ERRNO,T4TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
                026430  104456                                                          TRAP    C$ERHRD
                026432  000625                                                          .WORD   405
                026434  030024                                                          .WORD   T4TSBA
                026436  016164                                                          .WORD   EXPREC
  4823
  4824
  4825 026440  004737  010354          35$:    JSR     PC,CKRAM        ;SEE IF DATA IN RAM IS CORRECT
  4826 026444  103404                          BCS     40$             ;BRANCH IF PACKET IN RAM IS CORRECT
  4830 026446                                  ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
                026446  104456                                                          TRAP    C$ERHRD
```

```
                                                                                    .WORD   406
                                                                                    .WORD   PKTRAM
                                                                                    .WORD   RAMERR
 4831
 4832 026456                            40$:    ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<
      026456                                                                                10000$:
      026456  104405                                                                          TRAP    C$ESEG
 4833
 4834 026460  012364  000006                    MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
 4835 026464  020327  003030                    CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
 4836 026470  103002                            BHIS    55$                     ;BRANCH IF ALL DATA DONE
 4837 026472  000137  026242                    JMP     5$                      ;BRANCH TILL BACK TO ZERO
 4838
 4839 026476                            55$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
      026476                                                                                L10051:
      026476  104403                                                                          TRAP    C$ESUB
 4840
 4841 026500  005737  002170                    TST     FATFLG                  ;ANY FATAL ERRORS ?
 4842 026504  001402                            BEQ     60$                     ;BRANCH IF NOT
 4843 026506  004737  017772                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 4844 026512                            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  #PR100                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                                                                                                                MOV     #PR100,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
```

```
 4892 026650  012702  102206                MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
 4893 026654  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 4894 026660  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 4895 026662  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 4896 026666  020201                 25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 4897 026670  001414                        BEQ     30$                     ;OKAY IF MATCH
 4898 026672  010100                        MOV     R1,R0                   ;DATA FROM TSSR
 4899 026674                                XOR     R2,R0                   ;FIND BITS IN ERROR
 4900 026704  020027  002000                CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
 4901 026710  001404                        BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
 4905 026712                                ERRHRD  ERRNO,T42REJ,PKTSSR     ;COMMAND NOT REJECTED
      026712  104456                                                                TRAP    C$ERHRD
      026714  000632                                                                .WORD   410
      026716  027451                                                                .WORD   T42REJ
      026720  011520                                                                .WORD   PKTSSR
 4906 026722                         30$:   CKLOOP                          ;LOOP ON ERROR ?
      026722  104406                                                                TRAP    C$CLP1
 4907 026724  032701  002000                BIT     #NBA,R1                 ;IS NBA BIT SET ?
 4908 026730  001004                        BNE     35$                     ;OKAY IF NBA SET
 4912 026732                                ERRHRD  ERRNO,T42NBA,PKTSSR     ;NBA NOT SET
      026732  104456                                                                TRAP    C$ERHRD
      026734  000633                                                                .WORD   411
      026736  027320                                                                .WORD   T42NBA
      026740  011520                                                                .WORD   PKTSSR
 4913 026742                         35$:
 4914 026742                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      026742                                                                10000$:
      026742  104405                                                                TRAP    C$ESEG
 4915
 4916 026744  062703  000004                ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
 4917 026750  020327  027320                CMP     R3,#T42DONE             ;COMPARE TO END OF TEST DATA
 4918 026754  103002                        BHIS    57$                     ;BRANCH IF ALL DATA TESTED
 4919 026756  000137  026526                JMP     5$                      ;BRANCH TILL BACK TO ZERO
 4920
 4921 026762                         57$:   ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      026762                                                                L10052:
      026762  104403                                                                TRAP    C$ESUB
```

```
                                        ;*
                                        ;
                                        ;TEST 4, SUBTEST 3
                                        ;
                                        ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
                                        ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
                                        ;
                                        ;-

                                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
                                                                                                T4.3:
                                                                                                        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

                                                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$ERDF
                                                                                                        .WORD   412
                                                                                                        .WORD   SFIERR
                                                                                                        .WORD   SFIMSG
                                        10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
                                                BIS     #1,T4DATA               ;MAKE ADDRESS ODD
                                                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$ERDF
                                                                                                        .WORD   413
                                                                                                        .WORD   T4SSR
                                                                                                        .WORD   PKTSSR
                                        15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                                        TRAP    C$CLP1
                                                ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
                                                                                                        TRAP    C$ESCAPE
                                                                                                        .WORD   L10053-.
                                                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
                                                BEQ     22$                     ;BRANCH IF NOT
                                                ERRHRD  ERRNO,T4INT,PKTSSR
                                                                                                        TRAP    C$ERHRD
                                                                                                        .WORD   414
                                                                                                        .WORD   T4INT
                                                                                                        .WORD   PKTSSR
                                        22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
                                                MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
                                                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
                                                BEQ     25$                     ;BRANCH IF NOT OFF-LINE
```

```
4970 027130  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
4971 027134  020201                  25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
4972 027136  001414                         BEQ     30$                     ;OKAY IF MATCH
4973 027140  010100                         MOV     R1,R0                   ;DATA FROM TSSR
4974 027142                                 XOR     R2,R0                   ;FIND BITS IN ERROR
4975 027152  020027  002000                CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
4976 027156  001404                         BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
4980 027160                                 ERRHRD  ERRNO,T44REJ,PKTSSR     ;COMMAND NOT REJECTED
     027160  104456                                                                 TRAP    C$ERHRD
     027162  000637                                                                 .WORD   415
     027164  027550                                                                 .WORD   T44REJ
     027166  011520                                                                 .WORD   PKTSSR
4981 027170                         30$:    CKLOOP                          ;LOOP ON ERROR ?
     027170  104406                                                                 TRAP    C$CLP1
4982 027172  032701  002000                BIT     #NBA,R1                 ;IS NBA BIT SET ?
4983 027176  001004                         BNE     35$                     ;OKAY IF NBA SET
4987 027200                                 ERRHRD  ERRNO,T42NBA,PKTSSR     ;NBA NOT SET
     027200  104456                                                                 TRAP    C$ERHRD
     027202  000640                                                                 .WORD   416
     027204  027320                                                                 .WORD   T42NBA
     027206  011520                                                                 .WORD   PKTSSR
4988
4989 027210                         35$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     027210                                                                     L10053:
     027210  104403                                                                 TRAP    C$ESUB
4990
4991 027212  005737  002170                TST     FATFLG                  ;ANY FATAL ERRORS ?
4992 027216  001402                         BEQ     60$                     ;BRANCH IF NOT
4993 027220  004737  017772                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4994 027224                         60$:
4995 027224                         EXIT    TST                             ;ALL DONE THIS TEST
     027224  104432                                                                 TRAP    C$EXIT
     027226  000756                                                                 .WORD   L10050-.
4996
4997                                ;+
4998                                ;LOCAL STORAGE FOR THIS TEST
4999                                ;-
5000
5002 027230                                 .BLKB   10-<.-TUV2A87>
5004 027240                         T4PACKET:                               ;COMMAND PACKET FOR TEST
5005 027240  100004                         .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
5006 027242  027250                         .WORD   T4DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
5007 027244  000000                         .WORD   0
5008 027246  000010                         .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
5009
5010 027250                         T4DATA:                                 ;CHARACTERISTICS DATA BLOCK
5011 027250  027264                         .WORD   T4BFR                   ;ADDRESS OF MESSAGE BUFFER
5012 027252  000000                         .WORD   0
5013 027254  000016                         .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
5014 027256  000000                         .WORD   0
5015 027260                         T4SP:
5016
5017 027260  000000  000000                .WORD   0,0                     ;SPACE
5018 027264                         T4BFR:  .BLKW   8.                      ;MESSAGE BUFFER
5019
5020                                ;+
5021                                ;
```

```
                                        ;TEST DATA FOR SUBTEST TWO
                                        ;
                                        ;DATA HAS FORMAT:
                                        ;
                                        ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
                                        ;       2ND WORD        BITS TO SET FOR TEST
                                        ;
                                        ;-

027304                                  T42DATA:
027304  000000  037140                          .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
027310  000002  000001                          .WORD   2,BIT0
027314  000004  100100                          .WORD   4,BIT6!BIT15
        027320                          T42DONE=.


                                        ;*
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

027320      116     102     101  T42NBA: .ASCIZ  'NBA Not Set On Rejected WRITE CHARACTERISTICS'
027376      127     122     111  T4NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
027451      127     122     111  T42REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
027550      127     122     111  T44REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
027646      103     157     156  T4SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
027735      125     156     145  T4INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
030024      111     156     143  T4TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
030107      127     162     151  TST4ID: .ASCIZ  'Write Characteristics'
                                            .EVEN
```

```
5053
5054                                    ;+
5055                                    ;
5056                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5057                                    ;
5058                                    ;-
5059
5060 030136                            T4REST:
5061 030136                                    SAVREG                          ;SAVE THE REGISTERS
5062 030142  012701  027240                     MOV     #T4PACKET,R1            ;START OF THE PACKET
5063 030146  012721  100004                     MOV     #1C0004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
5064 030152  012721  027250                     MOV     #T4DATA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
5065 030156  005021                             CLR     (R1)+                   ;EXTENDED ADDRESS
5066 030160  012721  000010                     MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
5067 030164  012721  027264                     MOV     #T4BFR,(R1)+            ;ADDRESS OF MESSAGE BUFFER
5068 030170  005021                             CLR     (R1)+
5069 030172  012721  000020                     MOV     #16.,(R1)+              ;LENGTH OF MESSAGE BUFFER
5070 030176  005021                             CLR     (R1)+
5071 030200  005011                             CLR     (R1)
5072 030202  000207                             RTS     PC                      ;RETURN
5073 030204                                     ENDTST
     030204                                                                                  L10050:
     030204  104401                                                                                TRAP    CSETST
5074
```

```
  5076                                                  .SBTTL  TEST  5: VOLUME CHECK
  5077
  5078
  5079                                          ;        THIS TEST VERIFIES THAT THE VOLUME CHECK (VCK) BIT, A FLAG HELD
  5080                                          ;        WITHIN THE M7454 AND APPEARING IN XSTO, IS SET BY INITIALIZE AND
  5081                                          ;        CLEARED BY EXECUTING A WRITE CHARACTERISTICS COMMAND WITH THE
  5082                                          ;        CVC BIT SET.  IT IS ALSO VERIFIED THAT A WRITE CHARACTERISTICS
  5083                                          ;        COMMAND WITH THE CVC BIT CLEAR DOES NOT AFFECT THE STATE OF THE
  5084                                          ;        VOLUME CHECK BIT.  THE ACTUAL FUNCTION OF VOLUME CHECK, THAT OF
  5085                                          ;        PREVENTING OR ALLOWING A TAPE MOTION COMMAND DEPENDING UPON
  5086                                          ;        WHETHER VOLUME CHECK IS SET OR CLEAR, IS NOT CHECKED BY THIS
  5087                                          ;        TEST;  THIS FUNCTIONALITY IS CHECKED IN THE INDIVIDUAL TESTS OF
  5088                                          ;        TAPE MOTION COMMANDS.
  5089                                          ;
  5090                                          ;        THE TEST PROCEEDS AS FOLLOWS:
  5091                                          ;
  5092                                          ;             1.  THE CONTROLLER IS INITIALIZED BY WRITING INTO THE TSSR.
  5093                                          ;
  5094                                          ;             2.  A WRITE CHARACTERISTICS COMMAND IS ISSUED (WITH CVC=0)
  5095                                          ;                 AND XSTO IN THE RETURNED MESSAGE BUFFER IS EXAMINED;
  5096                                          ;                 THE VCK BIT SHOULD BE CLEAR (0).
  5097                                          ;
  5098                                          ;             3.  THE PREVIOUS STEP IS REPEATED TO VERIFY THAT VCK DOES
  5099                                          ;                 NOT CHANGE (REMAINS AT 0).
  5100                                          ;
  5101                                          ;             4.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=1
  5102                                          ;                 AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
  5103                                          ;                 EXAMINED;  THE VCK BIT SHOULD BE CLEAR (0).
  5104                                          ;
  5105                                          ;             5.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=0
  5106                                          ;                 AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
  5107                                          ;                 EXAMINED;  THE VCK BIT SHOULD REMAIN CLEAR (0).
  5108                                          ;
  5109                                          ;
  5110 030206                                          BGNTST
       030206                                                                                T5::
  5111 030206   005037  002170                          CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
  5112 030212   012737  005672  002146                  MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
  5113 030220   005037  003100                          CLR     KTFLG           ;HOLD OFF KT11
  5118 030224   012700  031377                          MOV     #TST5ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
  5119 030230   004737  017226                          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
  5120 030234   012737  000002  002164                  MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
  5121 030242                                  T5LOOP:
  5122
  5123 030242   012704  030710                          MOV     #T5PACKET,R4            ;PACKET FOR WRITE CHARACTERISTICS
  5124 030246   012702  030732                          MOV     #T5BFR,R2               ;ADDRESS OF THE MESSAGE BUFFER
  5125 030252   012762  052525  000006                  MOV     #052525,XSTO(R2)        ;SET XSTATO TO KNOWN VALUE
  5126 030260   004737  016464                  5$:     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
  5127 030264   103405                                  BCS     10$                     ;BR IF SOFT INIT = OK
  5131 030266   010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  5132 030270                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                                                                                                 TRAP    C$ERDF
       030270   104455                                                                           .WORD   501
       030272   000765                                                                           .WORD   SFIERR
       030274   003550                                                                           .WORD   SFIMSG
       030276   011506
  5133 030300   042714  040000                  10$:    BIC     #BIT14,(R4)             ;CLEAR THE CVC BIT
  5134 030304   010465  177776                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
```

```
5135 030310  004737  017054                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5136 030314  103405                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
5137 030316  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5141 030320                                ERRDF   ERRNO,T5SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     030320  104455                                                                TRAP    C$ERDF
     030322  000766                                                                .WORD   502
     030324  031211                                                                .WORD   T5SSR
     030326  011520                                                                .WORD   PKTSSR
5142 030330                         15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     030330  104406                                                                TRAP    C$CLP1
5143 030332                                ESCAPE  TST                     ;EXIT IF FATAL ERROR
     030332  104410                                                                TRAP    C$ESCAPE
     030334  001060                                                                .WORD   L10054-.
5144 030336  016203  000006                MOV     XSTO(R2),R3             ;STORE STATUS FOR A WHILE
5145 030342  020327  052525                CMP     R3,#052525              ;CHECK FOR XSTATO OVER WRITTEN (GOOD!)
5146 030346  001006                        BNE     20$                     ;BR, IF XSTATO HAS BEEN UPDATED
5147 030350  016501  000000                MOV     TSSR(R5),R1             ;PICK UP TSSR FOR ERROR PRINTOUT
5151 030354                                ERRHRD  ERRNO,T5NMSG,PKTSSR     ;'NO MESSAGE PACKET RETURNED'
     030354  104456                                                                TRAP    C$ERHRD
     030356  000767                                                                .WORD   503
     030360  031300                                                                .WORD   T5NMSG
     030362  011520                                                                .WORD   PKTSSR
5152 030364  032762  000020  000006  20$:   BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK CLEAR IN XSTO ?
5153 030372  001006                        BNE     23$                     ;OKAY IF VOLUME CHECK IS CLEARED
5157 030374  016501  000000                MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
5158 030400                                ERRHRD  ERRNO,T5VCK2,PKTMES     ;VOLUME CHECK NOT SET
     030400  104456                                                                TRAP    C$ERHRD
     030402  000770                                                                .WORD   504
     030404  031045                                                                .WORD   T5VCK2
     030406  011574                                                                .WORD   PKTMES
5159 030410                         23$:   CKLOOP                          ;LOOP ON ERROR ?
     030410  104406                                                                TRAP    C$CLP1
5160 030412  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
5161 030416  004737  017054                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5162 030422  103405                        BCS     25$                     ;BR IF CARRY SET (GOOD RETURN)
5163 030424  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5167 030426                                ERRDF   ERRNO,T5SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     030426  104455                                                                TRAP    C$ERDF
     030430  000771                                                                .WORD   505
     030432  031211                                                                .WORD   T5SSR
     030434  011520                                                                .WORD   PKTSSR
5168 030436                         25$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     030436  104406                                                                TRAP    C$CLP1
5169 030440                                ESCAPE  TST                     ;EXIT IF FATAL ERROR
     030440  104410                                                                TRAP    C$ESCAPE
     030442  000752                                                                .WORD   L10054-.
5170 030444  026203  000006                CMP     XSTO(R2),R3             ;THE XSTO SHOULD NOT HAVE CHANGED
5171 030450  001406                        BEQ     27$                     ;OKAY IF VOLUME CHECK IS SET
5175 030452  016501  000000                MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
5176 030456                                ERRHRD  ERRNO,T5NVCK,PKTMES     ;VOLUME CHECK NOT SET
     030456  104456                                                                TRAP    C$ERHRD
     030460  000772                                                                .WORD   506
     030462  031121                                                                .WORD   T5NVCK
     030464  011574                                                                .WORD   PKTMES
5177 030466                         27$:   CKLOOP                          ;LOOP ON ERROR ?
     030466  104406                                                                TRAP    C$CLP1
5178 030470  032762  000020  000006  30$:   BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK SET IN XSTO ?
```

```
5179 030476  001006                        BNE     33$                 ;OKAY IF VOLUME CHECK IS SET
5183 030500  016501  000000                MOV     TSSR(R5),R1         ;CONTENTS OF TSSR FOR ERROR REPORT
5184 030504                                ERRHRD  ERRNO,T5VCK2,PKTMES ;VOLUME CHECK NOT SET
     030504  104456                                                    TRAP    C$ERHRD
     030506  000773                                                    .WORD   507
     030510  031045                                                    .WORD   T5VCK2
     030512  011574                                                    .WORD   PKTMES
5185 030514                         33$:   CKLOOP                      ;LOOP ON ERROR ?
     030514  104406                                                    TRAP    C$CLP1
5186 030516  052714  040000                BIS     #BIT14,(R4)         ;SET THE CVC BIT
5187 030522  010465  177776                MOV     R4,TSDB(R5)         ;SET THE PACKET ADDRESS FOR WRITE CHAR
5188 030526  004737  017054                JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
5189 030532  103405                        BCS     35$                 ;BR IF CARRY SET (GOOD RETURN)
5190 030534  010001                        MOV     R0,R1               ;SAVE CONTENTS OF TSSR
5194 030536                                ERRDF   ERRNO,T5SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     030536  104455                                                    TRAP    C$ERDF
     030540  000774                                                    .WORD   508
     030542  031211                                                    .WORD   T5SSR
     030544  011520                                                    .WORD   PKTSSR
5195 030546                         35$:   CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
     030546  104406                                                    TRAP    C$CLP1
5196 030550                                ESCAPE  TST                 ;EXIT IF FATAL ERROR
     030550  104410                                                    TRAP    C$ESCAPE
     030552  000642                                                    .WORD   L10054-.
5197 030554  032762  000020  000006        BIT     #XSOVCK,XSTO(R2)    ;IS VOLUME CHECK CLEAR IN XSTO ?
5198 030562  001406                        BEQ     40$                 ;OKAY IF VOLUME CHECK IS CLEARED
5202 030564  016501  000000                MOV     TSSR(R5),R1         ;CONTENTS OF TSSR FOR ERROR REPORT
5203 030570                                ERRHRD  ERRNO,T5VCK,PKTMES  ;VOLUME CHECK NOT CLEARED
     030570  104456                                                    TRAP    C$ERHRD
     030572  000775                                                    .WORD   509
     030574  030752                                                    .WORD   T5VCK
     030576  011574                                                    .WORD   PKTMES
5204 030600                         40$:   CKLOOP                      ;LOOP ON ERROR ?
     030600  104406                                                    TRAP    C$CLP1
5205 030602  042714  040000                BIC     #BIT14,(R4)         ;CLEAR THE CVC BIT
5206 030606  010465  177776                MOV     R4,TSDB(R5)         ;SET THE PACKET ADDRESS FOR WRITE CHAR
5207 030612  004737  017054                JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
5208 030616  103405                        BCS     45$                 ;BR IF CARRY SET (GOOD RETURN)
5209 030620  010001                        MOV     R0,R1               ;SAVE CONTENTS OF TSSR
5213 030622                                ERRDF   ERRNO,T5SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     030622  104455                                                    TRAP    C$ERDF
     030624  000776                                                    .WORD   510
     030626  031211                                                    .WORD   T5SSR
     030630  011520                                                    .WORD   PKTSSR
5214 030632                         45$:   CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
     030632  104406                                                    TRAP    C$CLP1
5215 030634                                ESCAPE  TST                 ;EXIT IF FATAL ERROR
     030634  104410                                                    TRAP    C$ESCAPE
     030636  000556                                                    .WORD   L10054-.
5216 030640  032762  000020  000006        BIT     #XSOVCK,XSTO(R2)    ;IS VOLUME CHECK CLEAR IN XSTO ?
5217 030646  001406                        BEQ     50$                 ;OKAY IF VOLUME CHECK IS CLEARED
5221 030650  016501  000000                MOV     TSSR(R5),R1         ;CONTENTS OF TSSR FOR ERROR REPORT
5222 030654                                ERRHRD  ERRNO,T5NVCK,PKTMES ;VOLUME CHECK NOT CLEARED
     030654  104456                                                    TRAP    C$ERHRD
     030656  000777                                                    .WORD   511
     030660  031121                                                    .WORD   T5NVCK
     030662  011574                                                    .WORD   PKTMES
```

TEST 5: VOLUME CHECK

```
5223 030664                            50$:    CKLOOP                          ;LOOP ON ERROR ?
     030664  104406                                                                    TRAP    C$CLP1
5224 030666  004737  017174            60$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
5225 030672  103002                            BCC     62$                     ;BRANCH IF NOT
5226 030674  000137  030242                    JMP     T5LOOP                  ;LOOP UNTIL COUNT EXPIRED
5227 030700                            62$:    EXIT    TST                     ;ALL DONE THIS TEST
     030700  104432                                                                    TRAP    C$EXIT
     030702  000512                                                                    .WORD   L10054-.
5228
5229                                   ;+
5230                                   ;LOCAL STORAGE FOR THIS TEST
5231                                   ;-
5232
5234 030704                                    .BLKB   10-<.-TUV2A&7>
5236 030710                            T5PACKET:                               ;COMMAND PACKET FOR TEST
5237 030710  100004                            .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND
5238 030712  030720                            .WORD   T5DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
5239 030714  000000                            .WORD   0
5240 030716  000010                            .WORD   10                      ;STARTING VALUE OF COUNTER
5241
5242 030720                            T5DATA:                                 ;CHARACTERISTICS DATA BLOCK
5243 030720  030732                            .WORD   T5BFR                   ;ADDRESS OF MESSAGE BUFFER
5244 030722  000000                            .WORD   0
5245 030724  000020                            .WORD   16.                     ;LENGTH OF MESSAGE BUFFER
5246 030726  000000  000000                    .WORD   0,0
5247
5248 030732                            T5BFR:  .BLKW   8.                      ;MESSAGE BUFFER
5249
5250
5251                                   ;+
5252                                   ;LOCAL TEXT MESSAGES FOR TEST
5253                                   ;-
5254
5255 030752     126     103     113   T5VCK:  .ASCIZ  'VCK Bit NOT Cleared After WRITE CHARACTERISTICS With CVC=1'
5256 031045     126     103     113   T5VCK2: .ASCIZ  'VCK Bit NOT Set After INITIALIZE With CVC=0'
5257 031121     126     103     113   T5NVCK: .ASCIZ  'VCK Bit Modified After WRITE CHARACTERISTICS With CVC=0'
5258 031211     103     157     156   T5SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Characteristics'
5259 031300     116     157     040   T5NMSG: .ASCIZ  'No Message Packet Returned To Host After WRITE CHARACTERISTICS'
5260 031377     126     157     154   TST5ID: .ASCIZ  'Volume Check'
5261                                           .EVEN
5262 031414                                    ENDTST
     031414                                                                 L10054:
     031414  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 XST0
                                ;       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 XST0 IS 0.
                                ;
031416                          BGNTST
031416                                                                  T6::
031416  005037  002170          CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
031422  012737  005672  002146  MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
031430  005037  003100          CLR     KTFLG           ;HOLD OFF KT11
031434  012700  033371          MOV     #TST6ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
031440  004737  017226          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
031444  012737  000002  002164  MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
031452                  T6LOOP:
031452                          BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                        T6.1:
031452                                                                  TRAP    C$BSUB
031452  104402
                                JSR     PC,T6REST               ;SET PACKET TO INITIAL VALUES
031454  004737  033416          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
031460                                                                  MOV     #PRI00,R0
031460  012700  000000                                                  TRAP    C$SPRI
031464  104441
                                MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
031466  012703  002732          MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
031472  012704  032320          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
031476  012764  000010  000006  5$:
031504                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
031504                                                                  TRAP    C$BSEG
031504  104404
                                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
031506  004737  016464          BCS     10$                     ;BR IF SOFT INIT = OK
031512  103405                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
031514  010001                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
031516                                                                  TRAP    C$ERDF
031516  104455                                                          .WORD   601
031520  001131                                                          .WORD   SFIERR
031522  003550                                                          .WORD   SFIMSG
031524  011506
031526  005037  002170  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
031532  005037  002172          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
031536  010465  177776          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
031542  004737  017054          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
031546  103407                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
031550  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
5321 031552                            ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     031552  104455                                                                  TRAP    C$ERDF
     031554  001132                                                                  .WORD   602
     031556  033037                                                                  .WORD   T6SSR
     031560  011520                                                                  .WORD   PKTSSR
5322 031562  004737  017720            JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5323 031566                    15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     031566  104406                                                                  TRAP    C$CLP1
5324 031570                            ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     031570  104410                                                                  TRAP    C$ESCAPE
     031572  000056                                                                  .WORD   10000$-.
5325 031574  005737  002172            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
5326 031600  001004                    BNE     22$                     ;BRANCH IF YES
5330 031602                            ERRHRD  ERRNO,T6NINT,PKTSSR
     031602  104456                                                                  TRAP    C$ERHRD
     031604  001133                                                                  .WORD   603
     031606  033126                                                                  .WORD   T6NINT
     031610  011520                                                                  .WORD   PKTSSR
5331 031612  016501  000000    22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
5332 031616  012702  000200            MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
5333 031622  032701  000100            BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
5334 031626  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
5335 031630  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
5336 031634  020201    25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
5337 031636  001404                    BEQ     30$                     ;OKAY IF MATCH
5341 031640                            ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT ZERO
     031640  104456                                                                  TRAP    C$ERHRD
     031642  001134                                                                  .WORD   604
     031644  032376                                                                  .WORD   T6NBA
     031646  011520                                                                  .WORD   PKTSSR
5342 031650            30$:
5343 031650                            ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     031650                                                                     10000$:
     031650  104405                                                                  TRAP    C$ESEG
5344
5345 031652  012364  000006            MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
5346 031656  020327  003030            CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
5347 031662  103002                    BHIS    55$                     ;BRANCH IF ALL DATA DONE
5348 031664  000137  031504            JMP     5$                      ;BRANCH TILL BACK TO ZERO
5349
5350 031670            55$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     031670                                                                     L10056:
     031670  104403                                                                  TRAP    C$ESUB
5351
5352 031672  005737  002170            TST     FATFLG                  ;ANY FATAL ERRORS ?
5353 031676  001402                    BEQ     70$                     ;BRANCH IF NOT
5354 031700  004737  017772            JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
5355 031704            70$:
```

```
5357                                    ;+
5358                                    ;
5359                                    ;TEST 6, SUBTEST 2
5360                                    ;
5361                                    ;CHECK THAT UNUSED BITS BEING SET CAUSES
5362                                    ;WRITE CHARACTERISTICS COMMAND TO BE REJECTED
5363                                    ;
5364                                    ;-
5365
5366 031704                                BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      031704                                                                              T6.2:
      031704  104402                                                                        TRAP    C$BSUB
5367
5368 031706                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      031706  012700  000000                                                              MOV     #PRI00,R0
      031712  104441                                                                     TRAP    C$SPRI
5369 031714  012703  032362                MOV     #T62DATA,R3             ;START OF TEST DATA FOR SUBTEST
5370 031720  012704  032320          5$:   MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
5371 031724  004737  033416                JSR     PC,T6REST               ;RESTORE PACKET TO STARTING VALUES
5372
5373 031730                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      031730  104404                                                                      TRAP    C$BSEG
5374
5375 031732  004737  016464                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
5376 031736  103405                      BCS     10$                     ;BR IF SOFT INIT = OK
5380 031740  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5381 031742                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      031742  104455                                                                     TRAP    C$ERDF
      031744  001135                                                                     .WORD   605
      031746  003550                                                                     .WORD   SFIERR
      031750  011506                                                                     .WORD   SFIMSG
5382 031752  005037  002172          10$:  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
5383 031756  010400                      MOV     R4,R0                   ;START OF THE COMMAND PACKET
5384 031760  061300                      ADD     (R3),R0                 ;OFFSET TO THE DATA WORD TO TEST
5385 031762  056310  000002                BIS     2(R3),(R0)              ;SET THE DATA BITS TO BE TESTED
5386 031766  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
5387 031772  004737  016740                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
5388 031776  103405                      BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
5389 032000  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5393 032002                                ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      032002  104455                                                                     TRAP    C$ERDF
      032004  001136                                                                     .WORD   606
      032006  033037                                                                     .WORD   T6SSR
      032010  011520                                                                     .WORD   PKTSSR
5394 032012                          15$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      032012  104406                                                                     TRAP    C$CLP1
5395 032014                                ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
      032014  104410                                                                     TRAP    C$ESCAPE
      032016  000056                                                                     .WORD   10000$-.
5396 032020  005737  002172                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
5397 032024  001004                      BNE     22$                     ;BRANCH IF YES
5401 032026                                ERRHRD  ERRNO,T6NINT,PKTSSR
      032026  104456                                                                     TRAP    C$ERHRD
      032030  001137                                                                     .WORD   607
      032032  033126                                                                     .WORD   T6NINT
      032034  011520                                                                     .WORD   PKTSSR
5402 032036  016501  000000          22$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

TEST 6: COMPLETION INTERRUPT

```
5403 032042  012702  102206                MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
5404 032046  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
5405 032052  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
5406 032054  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
5407 032060  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
5408 032062  001404                        BEQ     30$                     ;OKAY IF MATCH
5412 032064                                ERRHRD  ERRNO,T62REJ,PKTSSR     ;COMMAND NOT REJECTED
     032064  104456                                                                TRAP    C$ERHRD
     032066  001140                                                                .WORD   608
     032070  032451                                                                .WORD   T62REJ
     032072  011520                                                                .WORD   PKTSSR
5413 032074                        30$:
5414 032074                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     032074                                                                   10000$:
     032074  104405                                                                TRAP    C$ESEG
5415
5416 032076  062703  000004                ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
5417 032102  020327  032376                CMP     R3,#T62DONE             ;COMPARE TO END OF TEST DATA
5418 032106  103002                        BHIS    59$                     ;BRANCH IF ALL DATA TESTED
5419 032110  000137  031720                JMP     5$                      ;BRANCH TILL BACK TO ZERO
5420
5421 032114                        59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     032114                                                                   L10057:
     032114  104403                                                                TRAP    C$ESUB
```

```
                                        ;+
                                        ;TEST 6, SUBTEST 3
                                        ;
                                        ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
                                        ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
                                        ;-
5429 032116                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     032116                                                                                     T6.3:
     032116  104402                                                                             TRAP    C$BSUB
5430 032120                             SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     032120  012700  000000                                                                     MOV     #PRI00,R0
     032124  104441                                                                             TRAP    C$SPRI
5431 032126  012703  032362            MOV     #T62DATA,R3             ;START OF TEST DATA FOR SUBTEST
5432 032132  012704  032320            5$:     MOV     #T6PACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
5433 032136  004737  033416                    JSR     PC,T6REST       ;RESTORE PACKET TO STARTING VALUES
5434 032142  004737  016464                    JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
5435 032146  103405                           BCS     10$             ;BR IF SOFT INIT = OK
5439 032150  010001                           MOV     R0,R1           ;SAVE CONTENTS OF TSSR
5440 032152                             ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     032152  104455                                                                             TRAP    C$ERDF
     032154  001141                                                                             .WORD   609
     032156  003550                                                                             .WORD   SFIERR
     032160  011506                                                                             .WORD   SFIMSG
5441 032162  005037  002172            10$:    CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
5442 032166  052737  000001  032330            BIS     #1,T6DATA       ;MAKE ADDRESS ODD
5443 032174  010465  177776                    MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
5444 032200  004737  016740                    JSR     PC,WAITF        ;WAIT FOR SSR TO SET
5445 032204  103405                           BCS     15$             ;BR IF CARRY SET (GOOD RETURN)
5446 032206  010001                           MOV     R0,R1           ;SAVE CONTENTS OF TSSR
5450 032210                             ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     032210  104455                                                                             TRAP    C$ERDF
     032212  001142                                                                             .WORD   610
     032214  033037                                                                             .WORD   T6SSR
     032216  011520                                                                             .WORD   PKTSSR
5451 032220                             15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
     032220  104406                                                                             TRAP    C$CLP1
5452 032222                             ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     032222  104410                                                                             TRAP    C$ESCAPE
     032224  000056                                                                             .WORD   L10060-.
5453 032226  005737  002172                    TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
5454 032232  001004                           BNE     22$             ;BRANCH IF YES
5458 032234                             ERRHRD  ERRNO,T6NINT,PKTSSR
     032234  104456                                                                             TRAP    C$ERHRD
     032236  001143                                                                             .WORD   611
     032240  033126                                                                             .WORD   T6NINT
     032242  011520                                                                             .WORD   PKTSSR
5459 032244  016501  000000            22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
5460 032250  012702  102206                    MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
5461 032254  032701  000100                    BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
5462 032260  001402                           BEQ     25$             ;BRANCH IF NOT OFF-LINE
5463 032262  052702  000100                    BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
5464 032266  020201                    25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
5465 032270  001404                           BEQ     30$             ;OKAY IF MATCH
5469 032272                             ERRHRD  ERRNO,T64REJ,PKTSSR     ;COMMAND NOT REJECTED
     032272  104456                                                                             TRAP    C$ERHRD
     032274  001144                                                                             .WORD   612
     032276  032643                                                                             .WORD   T64REJ
```

```
        032300  011520                                                          .WORD   PKTSSR
5470    032302                          30$:
5471    032302                                  ENDSUB                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        032302                                                                  L10060:
        032302  104403                                                          TRAP    C$ESUB
5472    032304                                  EXIT    TST             ;ALL DONE THIS TEST
        032304  104432                                                          TRAP    C$EXIT
        032306  001162                                                          .WORD   L10055-.
5473                                    ;+
5474                                    ;LOCAL STORAGE FOR THIS TEST
5475                                    ;-
5477    032310                                  .BLKB   10-<.-TUV2A87>
5479    032320                          T6PACKET:                       ;COMMAND PACKET FOR TEST
5480    032320  100204                          .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
5481    032322  032330                          .WORD   T6DATA          ;ADDRESS OF CHARACTERISTICS BLOCK
5482    032324  000000                          .WORD   0
5483    032326  000010                          .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
5484    032330                          T6DATA:                         ;CHARACTERISTICS DATA BLOCK
5485    032330  032342                          .WORD   T6BFR           ;ADDRESS OF MESSAGE BUFFER
5486    032332  000000                          .WORD   0
5487    032334  000016                          .WORD   14.             ;LENGTH OF MESSAGE BUFFER
5488    032336  000000  000000                  .WORD   0,0
5489    032342                          T6BFR:  .BLKW   8.              ;MESSAGE BUFFER
5490                                    ;+
5491                                    ;TEST DATA FOR SUBTEST TWO
5492                                    ;DATA HAS FORMAT:
5493                                    ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
5494                                    ;       2ND WORD        BITS TO SET FOR TEST
5495                                    ;-
5496    032362                          T62DATA:
5497    032362  000000  036140                  .WORD   0,BIT5!BIT6!BIT6!BIT10!BIT11!BIT12!BIT13
5498    032366  000002  000001                  .WORD   2,BIT0
5499    032372  000004  100100                  .WORD   4,BIT6!BIT15
5500            032376                  T62DONE=.
5501                                    ;+
5502                                    ;LOCAL TEXT MESSAGES FOR TEST
5503                                    ;-
5504    032376      127     122     111 T6NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
5505    032451      127     122     111 T62REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
5506    032550      127     122     111 T63REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
5507    032643      127     122     111 T64REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
5508    032741      127     122     111 T65REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
5509    033037      103     157     156 T6SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
5510    033126      105     170     160 T6NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
5511    033217      125     156     145 T6INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
5512    033306      111     156     143 T6TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
5513    033371      103     157     155 TST6ID: .ASCIZ  'Completion Interrupt'
5514                                            .EVEN
```

TEST 6: COMPLETION INTERRUPT

```
5516                                    ;+
5517                                    ;
5518                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5519                                    ;
5520                                    ;-
5521
5522 033416                            T6REST:
5523 033416                                    SAVREG                          ;SAVE THE REGISTERS
5524 033422  012701  032320                     MOV     #T6PACKET,R1            ;START OF THE PACKET
5525 033426  012721  100204                     MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
5526 033432  012721  032330                     MOV     #T6DATA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
5527 033436  005021                             CLR     (R1)+                   ;EXTENDED ADDRESS
5528 033440  012721  000010                     MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
5529 033444  012721  032342                     MOV     #T6BFR,(R1)+            ;ADDRESS OF MESSAGE BUFFER
5530 033450  005021                             CLR     (R1)+
5531 033452  012721  000016                     MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
5532 033456  005021                             CLR     (R1)+
5533 033460  005011                             CLR     (R1)
5534 033462  005037  032342                     CLR     T6BFR                   ;CLEAR 1ST LOC IN MESSAGE BUFFER
5535 033466  000207                             RTS     PC                      ;RETURN
5536 033470                                     ENDTST
     033470                                                                             L10055:
     033470  104401                                                                            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
                                ;
                                ;-
033472                                  BGNTST                                                  T7::
033472
033472  005037  002170                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
033476  012737  005672  002146          MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
033504  005037  003100                  CLR     KTFLG           ;HOLD OFF KT11
033510  012700  036527                  MOV     #TST7ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
033514  004737  017226                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
033520  012737  000002  002164          MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
033526                          T7LOOP:
                                ;
033526                                  BGNSUB                          ;/////////// BEGIN SUBTEST ///////////
033526                                                                                  T7.1:
033526  104402                                                                          TRAP    C$BSUB
                                ;
033530  004737  036556                  JSR     PC,T7RST                ;SET PACKET TO INITIAL VALUES
033534                                  SETPRI  #PR100                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
033534  012700  000000                                                                  MOV     #PR100,R0
033540  104441                                                                          TRAP    C$SPRI
033542  012704  035630                  MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
033546  012764  000010  000006          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
033554                          5$:
033554                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
033554  104404                                                                          TRAP    C$BSEG
                                ;
033556  004737  016464                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
033562  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
033564  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
033566                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
033566  104455                                                                          TRAP    C$ERDF
033570  001275                                                                          .WORD   701
033572  003550                                                                          .WORD   SFIERR
033574  011506                                                                          .WORD   SFIMSG
033576  005037  002170          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
033602  005037  002172                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
033606  010465  177776                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
033612  004737  017054                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
033616  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
033620  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
033622                                  ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
033622  104455                                                                          TRAP    C$ERDF
033624  001276                                                                          .WORD   702
033626  036260                                                                          .WORD   T7SSR
```

```
     033630  011520                                                          .WORD   PKTSSR
5592 033632  004737  017720             JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5593 033636                     15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     033636  104406                                                          TRAP    C$CLP1
5594 033640                             ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     033640  104410                                                          TRAP    C$ESCAPE
     033642  000056                                                          .WORD   10000$-.
5595 033644  005737  002172             TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
5596 033650  001004                     BNE     22$                     ;BRANCH IF YES
5600 033652                             ERRHRD  ERRNO,T7NINT,PKTSSR
     033652  104456                                                          TRAP    C$ERHRD
     033654  001277                                                          .WORD   703
     033656  036347                                                          .WORD   T7NINT
     033660  011520                                                          .WORD   PKTSSR
5601 033662  016501  000000     22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
5602 033666  012702  000200             MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
5603 033672  032701  000100             BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
5604 033676  001402                     BEQ     25$                     ;BRANCH IF NOT OFF-LINE
5605 033700  052702  000100             BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
5606 033704  020201                 25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
5607 033706  001404                     BEQ     30$                     ;OKAY IF MATCH
5611 033710                             ERRHRD  ERRNO,T7SSR,PKTSSR      ;NBA NOT ZERO
     033710  104456                                                          TRAP    C$ERHRD
     033712  001300                                                          .WORD   704
     033714  036260                                                          .WORD   T7SSR
     033716  011520                                                          .WORD   PKTSSR
5612 033720                     30$:
5613 033720                             ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
     033720                                                          10000$:
     033720  104405                                                          TRAP    C$ESEG
5614 033722                             BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     033722  104404                                                          TRAP    C$BSEG
5615
5616 033724  005037  002172             CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
5617 033730  012737  025252  035652     MOV     #025252,T7BFR           ;WIPE OUT MESSAGE BUFFER AREA
5618 033736  012714  100212             MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
5619 033742  010465  177776             MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
5620 033746  004737  017054             JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5621 033752  103407                     BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
5622 033754  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5626 033756                             ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     033756  104455                                                          TRAP    C$ERDF
     033760  001301                                                          .WORD   705
     033762  036260                                                          .WORD   T7SSR
     033764  011520                                                          .WORD   PKTSSR
5627 033766  004737  017720             JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5628 033772                     45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     033772  104406                                                          TRAP    C$CLP1
5629 033774  005737  002172             TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
5630 034000  001404                     BEQ     52$                     ;BRANCH IF NO
5634 034002                             ERRHRD  ERRNO,T7INT,PKTSSR
     034002  104456                                                          TRAP    C$ERHRD
     034004  001302                                                          .WORD   706
     034006  036440                                                          .WORD   T7INT
     034010  011520                                                          .WORD   PKTSSR
5635 034012  016501  000000     52$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
5636 034016  012702  000200             MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
```

```
 5637 034022  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 5638 034026  001402                        BEQ     55$                     ;BRANCH IF NOT OFF-LINE
 5639 034030  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 5640 034034  020201                  55$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 5641 034036  001404                        BEQ     60$                     ;OKAY IF MATCH
 5645 034040                                ERRHRD  ERRNO,T7SSR,PKTSSR      ;"TSSR INCORRECT AFTER WRT CHARA."
      034040  104456                                                                TRAP    C$ERHRD
      034042  001303                                                                .WORD   707
      034044  036260                                                                .WORD   T7SSR
      034046  011520                                                                .WORD   PKTSSR
 5646 034050                                60$:
 5647 034050  013701  035652                MOV     T7BFR,R1                ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
 5648 034054  012702  025252                MOV     #025252,R2              ;SET UP EXPECTED DATA
 5649 034060  020102                        CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
 5650 034062  001404                        BEQ     70$                     ;BR, IF OK (EQUAL)
 5654 034064                                ERRHRD  ERRNO,T7MBF,EXPREC      ;MESSAGE BUFFER WAS MODIFIED
      034064  104456                                                                TRAP    C$ERHRD
      034066  001304                                                                .WORD   708
      034070  035734                                                                .WORD   T7MBF
      034072  016164                                                                .WORD   EXPREC
 5655 034074                                70$:
 5656 034074  005737  002170                TST     FATFLG                  ;ANY FATAL ERRORS
 5657 034100  001403                        BEQ     80$                     ;BR, IF NO FATAL ERRORS
 5658 034102  004737  017772                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 5659 034106                                ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<
      034106                                                                10001$:
      034106  104405                                                                TRAP    C$ESEG
 5660 034110                                80$:
 5661 034110                                ENDSUB                          ;////////////// END SUBTEST ////////////
      034110                                                                L10062:
      034110  104403                                                                TRAP    C$ESUB
 5662
```

```
                                        ;+
                                        ;TEST 7, SUBTEST 2
                                        ;
                                        ;CHECKS THAT THE MESSAGE BUFFER RELEASE COMMAND WORKS
                                        ;PROPERLY AND THAT THERE IS AN INTERRUPT IF THE "IE"
                                        ;BIT IS SET IN THE COMMAND PACKET AND THE "ERI" BIT
                                        ;IS SET IN THE CHARACTERISTICS DATA PACKET
                                        ;
                                        ;-
034112                                          BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                                                        T7.2:
034112                                                                                                          TRAP    C$BSUB
034112  104402
                                                JSR     PC,T7RST                ;SET PACKET TO INITIAL VALUES
034114  004737  036556                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
034120                                                                                                          MOV     #PRI00,R0
034120  012700  000000                                                                                        TRAP    C$SPRI
034124  104441
                                                MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
034126  012704  035630                          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
034132  012764  000010  000006
                                        5$:
034140                                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
034140                                                                                                          TRAP    C$BSEG
034140  104404
                                                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
034142  004737  016464                          BCS     10$                     ;BR IF SOFT INIT = OK
034146  103405                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
034150  010001                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
034152                                                                                                          TRAP    C$ERDF
034152  104455                                                                                                  .WORD   709
034154  001305                                                                                                  .WORD   SFIERR
034156  003550                                                                                                  .WORD   SFIMSG
034160  011506
                                        10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
034162  005037  002170                          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
034166  005037  002172                          MOV     #000020,T7DATA+6        ;SET ERI IN CHARACTERISTICS DATA
034172  012737  000020  035646                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
034200  010465  177776                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
034204  004737  017054                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
034210  103407                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
034212  010001                                  ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
034214                                                                                                          TRAP    C$ERDF
034214  104455                                                                                                  .WORD   710
034216  001306                                                                                                  .WORD   T7SSR
034220  036260                                                                                                  .WORD   PKTSSR
034222  011520
                                                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
034224  004737  017720                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
034230                                                                                                          TRAP    C$CLP1
034230  104406
                                                ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
034232                                                                                                          TRAP    C$ESCAPE
034232  104410                                                                                                  .WORD   10000$-.
034234  000056
                                                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
034236  005737  002172                          BNE     22$                     ;BRANCH IF YES
034242  001004                                  ERRHRD  ERRNO,T7NINT,PKTSSR
034244                                                                                                          TRAP    C$ERHRD
034244  104456                                                                                                  .WORD   711
034246  001307                                                                                                  .WORD   T7NINT
034250  036347                                                                                                  .WORD   PKTSSR
034252  011520
                                        22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
034254  016501  000000
```

```
5710 034260  012702  000200                    MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
5711 034264  032701  000100                    BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
5712 034270  001402                            BEQ     25$                     ;BRANCH IF NOT OFF-LINE
5713 034272  052702  000100                    BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
5714 034276  020201                    25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
5715 034300  001404                            BEQ     30$                     ;OKAY IF MATCH
5719 034302                                    ERRHRD  ERRNO,T7SSR,PKTSSR      ;NBA NOT ZERO
     034302  104456                                                                    TRAP    C$ERHRD
     034304  001310                                                                    .WORD   712
     034306  036260                                                                    .WORD   T7SSR
     034310  011520                                                                    .WORD   PKTSSR
5720 034312                            30$:
5721 034312                            ENDSEG                                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     034312                                                                                 10000$:
     034312  104405                                                                    TRAP    C$ESEG
5722 034314                            BGNSEG                                  ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     034314  104404                                                                    TRAP    C$BSEG
5723
5724 034316  005037  002172                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
5725 034322  012737  025252  035652            MOV     #025252,T7BFR           ;WIPE OUT MESSAGE BUFFER AREA
5726 034330  012714  100212                    MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
5727 034334  010465  177776                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
5728 034340  004737  017054                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5729 034344  103407                            BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
5730 034346  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5734 034350                                    ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     034350  104455                                                                    TRAP    C$ERDF
     034352  001311                                                                    .WORD   713
     034354  036260                                                                    .WORD   T7SSR
     034356  011520                                                                    .WORD   PKTSSR
5735 034360  004737  017720                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5736 034364                            45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034364  104406                                                                    TRAP    C$CLP1
5737 034366  005737  002172                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
5738 034372  001004                            BNE     52$                     ;BRANCH IF YES
5742 034374                                    ERRHRD  ERRNO,T7INT,PKTSSR
     034374  104456                                                                    TRAP    C$ERHRD
     034376  001312                                                                    .WORD   714
     034400  036440                                                                    .WORD   T7INT
     034402  011520                                                                    .WORD   PKTSSR
5743 034404  016501  000000            52$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
5744 034410  012702  000200                    MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
5745 034414  032701  000100                    BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
5746 034420  001402                            BEQ     55$                     ;BRANCH IF NOT OFF-LINE
5747 034422  052702  000100                    BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
5748 034426  020201                    55$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
5749 034430  001404                            BEQ     60$                     ;OKAY IF MATCH
5753 034432                                    ERRHRD  ERRNO,T7SSR,PKTSSR      ;NBA NOT SET
     034432  104456                                                                    TRAP    C$ERHRD
     034434  001313                                                                    .WORD   715
     034436  036260                                                                    .WORD   T7SSR
     034440  011520                                                                    .WORD   PKTSSR
5754 034442                            60$:
5755 034442  013701  035652                    MOV     T7BFR,R1                ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
5756 034446  012702  025252                    MOV     #025252,R2              ;SET UP EXPECTED DATA
5757 034452  020102                            CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
5758 034454  001404                            BEQ     70$                     ;BR, IF OK (EQUAL)
```

```
5762 034456                              ERRHRD  ERRNO,T7MBF,EXPREC      ;MESSAGE BUFFER WAS MODIFIED
     034456  104456                                                              TRAP    C$ERHRD
     034460  001314                                                              .WORD   716
     034462  035734                                                              .WORD   T7MBF
     034464  016164                                                              .WORD   EXPREC
5763
5764 034466                         70$:
5765 034466  005737  002170                 TST     FATFLG                  ;ANY FATAL ERRORS
5766 034472  001402                         BEQ     80$                     ;BR, IF NO FATAL ERRORS
5767 034474  004737  017772                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
5768 034500                         80$:
5769 034500                                 ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     034500                                                                        10001$:
     034500  104405                                                              TRAP    C$ESEG
5770 034502                                 ENDSUB                          ;////////////// END SUBTEST //////////////
     034502                                                                        L10063:
     034502  104403                                                              TRAP    C$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
                                        ;
                                        ;-
034504                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
034504                                                                                      T7.3:
034504  104402                                                                              TRAP    C$BSUB

034506  004737  036556                          JSR     PC,T7RST                ;SET PACKET TO INITIAL VALUES
034512                                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
034512  012700  000000                                                                      MOV     #PRI00,R0
034516  104441                                                                              TRAP    C$SPRI
034520  012704  035630                          MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
034524  012764  000010  000006                  MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
034532                                  5$:
034532                                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
034532  104404                                                                              TRAP    C$BSEG

034534  004737  016464                          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
034540  103405                                  BCS     10$                     ;BR IF SOFT INIT = OK
034542  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
034544                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
034544  104455                                                                              TRAP    C$ERDF
034546  001315                                                                              .WORD   717
034550  003550                                                                              .WORD   SFIERR
034552  011506                                                                              .WORD   SFIMSG

034554  005037  002170          10$:            CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
034560  005037  002172                          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
034564  010465  177776                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
034570  004737  017054                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
034574  103407                                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
034576  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
034600                                          ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
034600  104455                                                                              TRAP    C$ERDF
034602  001316                                                                              .WORD   718
034604  036260                                                                              .WORD   T7SSR
034606  011520                                                                              .WORD   PKTSSR
034610  004737  017720                          JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
034614                          15$:            CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
034614  104406                                                                              TRAP    C$CLP1
034616                                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
034616  104410                                                                              TRAP    C$ESCAPE
034620  000056                                                                              .WORD   10000$-.
034622  005737  002172                          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
034626  001004                                  BNE     22$                     ;BRANCH IF YES
034630                                          ERRHRD  ERRNO,T7NINT,PKTSSR
034630  104456                                                                              TRAP    C$ERHRD
034632  001317                                                                              .WORD   719
034634  036347                                                                              .WORD   T7NINT
034636  011520                                                                              .WORD   PKTSSR
034640  016501  000000          22$:            MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
034644  012702  000200                          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
034650  032701  000100                          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
```

```
5818 034654  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
5819 034656  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
5820 034662  020201                  25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
5821 034664  001404                        BEQ     30$                     ;OKAY IF MATCH
5825 034666                                ERRHRD  ERRNO,T7SSR,PKTSSR      ;NBA NOT ZERO
     034666  104456                                                                TRAP    C$ERHRD
     034670  001320                                                                .WORD   720
     034672  036260                                                                .WORD   T7SSR
     034674  011520                                                                .WORD   PKTSSR
5826 034676                             30$:
5827 034676                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     034676                                                                     10000$:
     034676  104405                                                               TRAP    C$ESEG
5828 034700                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     034700  104404                                                               TRAP    C$BSEG
5829 034702  004737  036556                JSR     PC,T7RST                ;RESET PACKETS AND COMMANDS
5830 034706  005037  002172                CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
5831 034712  012737  025252  035652        MOV     #025252,T7BFR           ;WIPE OUT MESSAGE BUFFER AREA
5832 034720  012714  100212                MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
5833 034724  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
5834 034730  004737  017054                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5835 034734  103407                        BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
5836 034736  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5840 034740                                ERRDF   ERRNO,T7SSRM,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     034740  104455                                                               TRAP    C$ERDF
     034742  001321                                                               .WORD   721
     034744  036170                                                               .WORD   T7SSRM
     034746  011520                                                               .WORD   PKTSSR
5841 034750  004737  017720                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5842 034754                          45$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034754  104406                                                               TRAP    C$CLP1
5843 034756  005737  002172                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
5844 034762  001404                        BEQ     52$                     ;BRANCH IF NO
5848 034764                                ERRHRD  ERRNO,T7INT,PKTSSR
     034764  104456                                                               TRAP    C$ERHRD
     034766  001322                                                               .WORD   722
     034770  036440                                                               .WORD   T7INT
     034772  011520                                                               .WORD   PKTSSR
5849 034774  016501  000000          52$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
5850 035000  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
5851 035004  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
5852 035010  001402                        BEQ     55$                     ;BRANCH IF NOT OFF-LINE
5853 035012  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
5854 035016  020201                  55$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
5855 035020  001404                        BEQ     60$                     ;OKAY IF MATCH
5859 035022                                ERRHRD  ERRNO,T7SSR,PKTSSR      ;NBA NOT SET
     035022  104456                                                               TRAP    C$ERHRD
     035024  001323                                                               .WORD   723
     035026  036260                                                               .WORD   T7SSR
     035030  011520                                                               .WORD   PKTSSR
5860 035032                             60$:
5861 035032  013701  035652                MOV     T7BFR,R1                ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
5862 035036  012702  025252                MOV     #025252,R2              ;SET UP EXPECTED DATA
5863 035042  020102                        CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
5864 035044  001404                        BEQ     70$                     ;BR, IF OK (EQUAL)
5868 035046                                ERRHRD  ERRNO,T7MBF,EXPREC      ;MESSAGE BUFFER WAS MODIFIED
     035046  104456                                                               TRAP    C$ERHRD
```

```
                035050  001324                                                          .WORD   724
                035052  035734                                                          .WORD   T7MBF
                035054  016164                                                          .WORD   EXPREC
    5869
    5870 035056                         70$:    CKLOOP                  ;LOOP ON ERROR IF FLAG SET
                035056  104406                                                          TRAP    C$CLP1
    5871
    5872 035060 005037  002172                  CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
    5873 035064 004737  036556                  JSR     PC,T7RST        ;RESET THE PACKETS AND COMMANDS
    5874 035070 042714  100000                  BIC     #100000,(R4)    ;CLEAR THE ACK BIT
    5875 035074 010465  177776                  MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
    5876 035100 004737  017054                  JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
    5877 035104 103407                          BCS     75$             ;BR IF CARRY SET (GOOD RETURN)
    5878 035106 010001                          MOV     R0,R1           ;SAVE CONTENTS OF TSSR
    5882 035110                                 ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                035110  104455                                                          TRAP    C$ERDF
                035112  001325                                                          .WORD   725
                035114  036260                                                          .WORD   T7SSR
                035116  011520                                                          .WORD   PKTSSR
    5883 035120 004737  017720                  JSR     PC,FATCHK       ;INC AND CHECK FOR MORE THAN 25 ERRORS
    5884 035124                         75$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                035124  104406                                                          TRAP    C$CLP1
    5885 035126                                 ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
                035126  104410                                                          TRAP    C$ESCAPE
                035130  000062                                                          .WORD   10001$-.
    5886 035132 005737  002172                  TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
    5887 035136 001006                          BNE     82$             ;BRANCH IF YES
    5891 035140 016500  000000                  MOV     TSSR(R5),R0     ;GET TSSR FOR ERROR REPORT
    5892 035144                                 ERRHRD  ERRNO,T7NINT,PKTSSR
                035144  104456                                                          TRAP    C$ERHRD
                035146  001326                                                          .WORD   726
                035150  036347                                                          .WORD   T7NINT
                035152  011520                                                          .WORD   PKTSSR
    5893 035154 016501  000000          82$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
    5894 035160 012702  000200                  MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
    5895 035164 032701  000100                  BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
    5896 035170 001402                          BEQ     85$             ;BRANCH IF NOT OFF-LINE
    5897 035172 052702  000100                  BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
    5898 035176 020201                  85$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
    5899 035200 001404                          BEQ     90$             ;OKAY IF MATCH
    5903 035202                                 ERRHRD  ERRNO,T7SSR,PKTSSR      ;NBA NOT ZERO
                035202  104456                                                          TRAP    C$ERHRD
                035204  001327                                                          .WORD   727
                035206  036260                                                          .WORD   T7SSR
                035210  011520                                                          .WORD   PKTSSR
    5904 035212                         90$:
    5905 035212                                 ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                035212                                                              10001$:
                035212  104405                                                          TRAP    C$ESEG
    5906 035214 005737  002170                  TST     FATFLG          ;ANY FATAL ERRORS
    5907 035220 001403                          BEQ     95$             ;BR, IF NO FATAL ERRORS
    5908 035222 004737  017772                  JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
    5909
    5910 035226                                 BGNSEG                  ;<<<<<<<<<<<<< BGN SEGMENT <<<<<<<<<<<<<
                035226  104404                                                          TRAP    C$BSEG
    5911 035230 005037  002172          95$:    CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
    5912 035234 004737  036556                  JSR     PC,T7RST        ;RESET THE PACKETS AND COMMANDS
```

```
 5913 035240  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 5914 035244  004737  017054                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 5915 035250  103407                        BCS     100$                    ;BR IF CARRY SET (GOOD RETURN)
 5916 035252  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5920 035254                                ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      035254  104455                                                                TRAP    C$ERDF
      035256  001330                                                                .WORD   728
      035260  036260                                                                .WORD   T7SSR
      035262  011520                                                                .WORD   PKTSSR
 5921 035264  004737  017720                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5922 035270                        100$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      035270  104406                                                                TRAP    C$CLP1
 5923 035272                                ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      035272  104410                                                                TRAP    C$ESCAPE
      035274  000062                                                                .WORD   10002$-.
 5924 035276  005737  002172                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 5925 035302  001006                        BNE     112$                    ;BRANCH IF YES
 5929 035304  016500  000000                MOV     TSSR(R5),R0             ;GET TSSR FOR ERROR REPORT
 5930 035310                                ERRHRD  ERRNO,T7NINT,PKTSSR
      035310  104456                                                                TRAP    C$ERHRD
      035312  001331                                                                .WORD   729
      035314  036347                                                                .WORD   T7NINT
      035316  011520                                                                .WORD   PKTSSR
 5931 035320  016501  000000        112$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 5932 035324  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 5933 035330  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 5934 035334  001402                        BEQ     125$                    ;BRANCH IF NOT OFF-LINE
 5935 035336  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 5936 035342  020201                125$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 5937 035344  001405                        BEQ     130$                    ;OKAY IF MATCH
 5941 035346                                ERRHRD  ERRNO,T7SSR,PKTSSR      ;NBA NOT ZERO
      035346  104456                                                                TRAP    C$ERHRD
      035350  001332                                                                .WORD   730
      035352  036260                                                                .WORD   T7SSR
      035354  011520                                                                .WORD   PKTSSR
 5942 035356                                ENDSEG                          ;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
      035356                                                                    10002$:
      035356  104405                                                                TRAP    C$ESEG
 5943 035360  005737  002170        130$:   TST     FATFLG                  ;CHECK FATAL FLAG FOR SET
 5944 035364  001402                        BEQ     140$                    ;BR, IF NOT SET (OK)
 5945 035366  004737  017772                JSR     PC,CKDROP               ;DROP DEVICE IF INDICATED
 5946 035372                        140$:
 5947 035372                                ENDSUB                          ;////////////// END SUBTEST ////////////
      035372                                                                    L10064:
      035372  104403                                                                TRAP    C$ESUB
 5948 035374                        30$:
```

```
                                        ;+
                                        ;TEST 7, SUBTEST 4
                                        ;
                                        ;CHECKS THAT THE REGISTER MODIFICATION REFUSED (RMR) BIT IN
                                        ;THE TSSR WILL BE SET IF A WRITE CHARACTERISTICS COMMAND
                                        ;BEING EXECUTED AND ANOTHER "WC" COMMAND IS ATTEMPTED
                                        ;
                                        ;-

035374                                          BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
035374                                                                                      T7.4:
035374  104402                                                                                    TRAP    C$BSUB

035376  004737  036630                          JSR     PC,T7RT2        ;SET SECOND PACKET UP
035402  004737  036556                          JSR     PC,T7RST        ;SET PACKET TO INITIAL VALUES
035406                                          SETPRI  #PR100          ;LOWER PRIORITY TO ALLOW INTERRUPTS
035406  012700  000000                                                                              MOV     #PR100,R0
035412  104441                                                                                      TRAP    C$SPRI
035414  012704  035630                          MOV     #T7PACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
035420  012703  035672                          MOV     #T7PKT,R3       ;GET THE ADDRESS OF 2ND CMD PACKET
035424  012764  000010  000006                  MOV     #8.,PKBCNT(R4)  ;START WITH MINIMUM ALLOWABLE VALUE
035432  012763  000010  000006                  MOV     #8.,PKBCNT(R3)  ;START WITH MINIMUM ALLOWABLE VALUE
035440                                  5$:
035440                                          BGNSEG                  ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
035440  104404                                                                                      TRAP    C$BSEG
035442  004737  016464                          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
035446  103405                                  BCS     10$             ;BR IF SOFT INIT = OK
035450  010001                                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
035452                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
035452  104455                                                                                      TRAP    C$ERDF
035454  001333                                                                                      .WORD   731
035456  003550                                                                                      .WORD   SFIERR
035460  011506                                                                                      .WORD   SFIMSG
035462  005037  002170                  10$:    CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
035466  005037  002172                          CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
035472  010465  177776                          MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
035476  010365  177776                          MOV     R3,TSDB(R5)     ;SECOND COMMAND PACKET
035502  004737  016740                          JSR     PC,WAITF        ;WAIT FOR SSR TO SET
035506  016501  000000                          MOV     TSSR(R5),R1     ;GET CONTENTS OF TSSR REGISTER
035512  032701  000200                          BIT     #SSR,R1         ;CHECK FOR SSR (TSSR) SET
035516  001006                                  BNE     15$             ;BR, IF SSR SET (GOOD)
035520                                          ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
035520  104455                                                                                      TRAP    C$ERDF
035522  001334                                                                                      .WORD   732
035524  036260                                                                                      .WORD   T7SSR
035526  011520                                                                                      .WORD   PKTSSR
035530  004737  017720                          JSR     PC,FATCHK       ;INC AND CHECK FOR MORE THAN 25 ERRORS
035534                                  15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
035534  104406                                                                                      TRAP    C$CLP1
035536                                          ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
035536  104410                                                                                      TRAP    C$ESCAPE
035540  000056                                                                                      .WORD   10000$-.
035542  005737  002172                          TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
035546  001004                                  BNE     22$             ;BRANCH IF YES

035550                                          ERRHRD  ERRNO,T7NINT,PKTSSR
```

```
                035550  104456                                                                  TRAP    C$ERHRD
                035552  001335                                                                  .WORD   733
                035554  036347                                                                  .WORD   T7NINT
                035556  011520                                                                  .WORD   PKTSSR
 6000 035560  016501  000000          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 6001 035564  012702  110200                  MOV     #SSR!RMR!SC,R2          ;EXPECTED CONTENTS OF TSSR
 6002 035570  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 6003 035574  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 6004 035576  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 6005 035602  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 6006 035604  001404                          BEQ     30$                     ;OKAY IF MATCH
 6010 035606                                  ERRHRD  ERRNO,T7SSR,PKTSSR      ;NBA NOT ZERO
                035606  104456                                                                  TRAP    C$ERHRD
                035610  001336                                                                  .WORD   734
                035612  036260                                                                  .WORD   T7SSR
                035614  011520                                                                  .WORD   PKTSSR
 6011 035616                          30$:
 6012 035616                                  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                035616                                                                  10000$:
                035616  104405                                                                  TRAP    C$ESEG
 6013 035620                                  ENDSUB                          ;////////////// END SUBTEST //////////////
                035620                                                                  L10065:
                035620  104403                                                                  TRAP    C$ESUB
 6014 035622                                  EXIT    TST                     ;ALL DONE WITH THIS TEST
                035622  104432                                                                  TRAP    C$EXIT
                035624  001056                                                                  .WORD   L10061-.
```

```
6016                                    ;+
6017                                    ;LOCAL STORAGE FOR THIS TEST
6018                                    ;-
6019
6021 035626                             T7PACKET:       .BLKB   10-<.-TUV2A87>
6023 035630                                             .WORD   100204          ;COMMAND PACKET FOR TEST
6024 035630  100204                                     .WORD   T7DATA          ;WRITE CHAR COMMAND, WITH IE, ACK
6025 035632  035640                                     .WORD   0               ;ADDRESS OF CHARACTERISTICS BLOCK
6026 035634  000000                                     .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
6027 035636  000010
6028
6029 035640                             T7DATA:                                 ;CHARACTERISTICS DATA BLOCK
6030 035640  035652                                     .WORD   T7BFR           ;ADDRESS OF MESSAGE BUFFER
6031 035642  000000                                     .WORD   0
6032 035644  000016                                     .WORD   14.             ;LENGTH OF MESSAGE BUFFER
6033 035646  000000  000000                             .WORD   0,0
6034
6035 035652                             T7BFR:  .BLKW   8.                      ;MESSAGE BUFFER
6036
6037                                    ;+
6038                                    ;
6039                                    ;TEST DATA FOR SUBTEST FOUR
6040                                    ;
6041 035672                             T7PKT:                                  ;COMMAND PACKET FOR TEST
6042 035672  100204                                     .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
6043 035674  035702                                     .WORD   T7DTA           ;ADDRESS OF CHARACTERISTICS BLOCK
6044 035676  000000                                     .WORD   0
6045 035700  000010                                     .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
6046
6047 035702                             T7DTA:                                  ;CHARACTERISTICS DATA BLOCK
6048 035702  035714                                     .WORD   T7BUFR          ;ADDRESS OF MESSAGE BUFFER
6049 035704  000000                                     .WORD   0
6050 035706  000016                                     .WORD   14.             ;LENGTH OF MESSAGE BUFFER
6051 035710  000000  000000                             .WORD   0,0
6052
6053 035714                             T7BUFR:.BLKW    8.                      ;MESSAGE BUFFER
6054
6055                                    ;+
6056                                    ;LOCAL TEXT MESSAGES FOR TEST
6057                                    ;-
6058
6059
6060 035734      115     145     163    T7MBF:  .ASCIZ  'Message Buffer Modified after MESSAGE BUFFER RELEASE Command'
6061 036031      116     102     101    T7NBA:  .ASCIZ  'NBA Not Clear After WRITE CHARACTERISTICS Command'
6062 036113      116     102     101    T7NNBA: .ASCIZ  'NBA Set After MESSAGE BUFFER RELEASE Command'
6063
6064 036170      103     157     156    T7SSRM: .ASCIZ  'Contents Of TSSR Incorrect After Message Buffer Release'
6065 036260      103     157     156    T7SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
6066 036347      105     170     160    T7NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
6067 036440      125     156     145    T7INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
6068 036527      102     141     163    TST7ID: .ASCIZ  'Basic Packet Protocol'
6069                                            .EVEN
6070
```

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

036556                                  T7RST:
036556                                          SAVREG                          ;SAVE THE REGISTERS
036562  012701  035630                          MOV     #T7PACKET,R1            ;START OF THE PACKET
036566  012721  100204                          MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
036572  012721  035640                          MOV     #T7DATA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
036576  005021                                  CLR     (R1)+                   ;EXTENDED ADDRESS
036600  012721  000010                          MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
036604  012721  035652                          MOV     #T7BFR,(R1)+            ;ADDRESS OF MESSAGE BUFFER
036610  005021                                  CLR     (R1)+
036612  012721  000016                          MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
036616  005021                                  CLR     (R1)+
036620  005011                                  CLR     (R1)
036622  005037  035652                          CLR     T7BFR                   ;CLEAR 1ST LOC IN MESSAGE BUFFER
036626  000207                                  RTS     PC                      ;RETURN
                                        :+
                                        :
                                        ;ROUTINE TO RESTORE COMMAND PACKET #2 TO START-UP (DEFAULT) VALUES
                                        :
                                        :-

036630                                  T7RT2:
036630                                          SAVREG                          ;SAVE THE REGISTERS
036634  012701  035672                          MOV     #T7PKT,R1               ;START OF THE PACKET
036640  012721  100204                          MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
036644  012721  035702                          MOV     #T7DTA,(R1)+            ;ADDRESS OF CHAR DATA BLOCK
036650  005021                                  CLR     (R1)+                   ;EXTENDED ADDRESS
036652  012721  000010                          MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
036656  012721  035714                          MOV     #T7BUFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
036662  005021                                  CLR     (R1)+
036664  012721  000016                          MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
036670  005021                                  CLR     (R1)+
036672  005011                                  CLR     (R1)
036674  005037  035714                          CLR     T7BUFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
036700  000207                                  RTS     PC                      ;RETURN
036702                                          ENDTST
036702                                                                  L10061:
036702  104401                                                                  TRAP    CSETST
```

```
                                        .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.
                                ;
                                ;-

036704                                  BGNTST
036704                                                                  T8::
036704  005037  002170                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
036710  012737  005672  002146          MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
036716  005037  003100                  CLR     KTFLG           ;HOLD OFF KT11
036722  012700  040262                  MOV     #TST8ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
036726  004737  017226                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
036732  012737  000002  002164          MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
036740                          T8LOOP:
036740                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
036740                                                                  T8.1:
036740  104402                                                                  TRAP    C$BSUB

036742                                  SETPRI  #PR100                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
036742  012700  000000                                                          MOV     #PR100,R0
036746  104441                                                                  TRAP    C$SPRI
036750  004737  016464                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
036754  103405                          BCS     3$                      ;BR IF SOFT INIT = OK
036756  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036760                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
036760  104455                                                                  TRAP    C$ERDF
036762  001441                                                                  .WORD   801
036764  003550                                                                  .WORD   SFIERR
036766  011506                                                                  .WORD   SFIMSG
036770                          3$:
036770  012704  037510                  MOV     #T8PK2,R4               ;WRITE CHARACTERISTICS PACKET
036774  004737  010152                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
037000  103404                          BCS     4$                      ;BR, IF COMMAND ISSUED OK
037002                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
037002  104456                                                                  TRAP    C$ERHRD
037004  001442                                                                  .WORD   802
037006  004754                                                                  .WORD   WRTMSG
037010  011506                                                                  .WORD   SFIMSG
037012                          4$:
037012  004737  040314                  JSR     PC,T8REST               ;SET UP PACKET FOR COMMAND
037016  012704  037440                  MOV     #T8PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
037022                          5$:
037022                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
037022  104404                                                                  TRAP    C$BSEG

037024  005037  002170          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
037030  005037  002172                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
037034  010465  177776                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
037040  004737  017054                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
037044  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
```

```
037046  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
037050                           ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
037050  104455                                                                   TRAP    C$ERDF
037052  001443                                                                   .WORD   803
037054  040004                                                                   .WORD   T8SSR
037056  011520                                                                   .WORD   PKTSSR
037060  004737  017720           JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
037064                   15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
037064  104406                                                                   TRAP    C$CLP1
037066                           ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
037066  104410                                                                   TRAP    C$ESCAPE
037070  000074                                                                   .WORD   10000$-.
037072  005737  002172           TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
037076  001004                   BNE     22$                     ;BRANCH IF YES
037100                           ERRHRD  ERRNO,T8NINT,PKTSSR
037100  104456                                                                   TRAP    C$ERHRD
037102  001444                                                                   .WORD   804
037104  040134                                                                   .WORD   T8NINT
037106  011520                                                                   .WORD   PKTSSR
037110  016501  000000   22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
037114  012702  000200           MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
037120  032701  000100           BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
037124  001402                   BEQ     25$                     ;BRANCH IF NOT OFF-LINE
037126  052702  000100           BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
037132  020201           25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
037134  001404                   BEQ     30$                     ;OKAY IF MATCH
037136                           ERRHRD  ERRNO,T8NBA,PKTSSR      ;NBA NOT ZERO
037136  104456                                                                   TRAP    C$ERHRD
037140  001445                                                                   .WORD   805
037142  037552                                                                   .WORD   T8NBA
037144  011520                                                                   .WORD   PKTSSR
037146                   30$:
037146  004737  010354   35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
037152  103405                   BCS     59$                     ;RAM OK GO ON
037154                           ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
037154  104456                                                                   TRAP    C$ERHRD
037156  001446                                                                   .WORD   806
037160  004643                                                                   .WORD   PKTRAM
037162  016200                                                                   .WORD   RAMERR
037164                           ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
037164                                                                   10000$:
037164  104405                                                                   TRAP    C$ESEG
037166                   59$:    ENDSUB                          ;\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\
037166                                                                   L10067:
037166  104403                                                                   TRAP    C$ESUB
037170  005737  002170           TST     FATFLG                  ;ANY FATAL ERRORS ?
037174  001402                   BEQ     60$                     ;BRANCH IF NOT
037176  004737  017772           JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
037202                   60$:
```

```
                                        ;+
                                        ;
                                        ;TEST 8, SUBTEST 2
                                        ;
                                        ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES
                                        ;INITIALIZE COMMAND TO BE REJECTED
                                        ;
                                        ;-

                                                BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                                                        T8.2:
037202  104402                                                                                                  TRAP    C$BSUB

                                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
037204  012700  000000                                                                                          MOV     #PRI00,R0
037210  104441                                                                                                  TRAP    C$SPRI
                                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
037212  104404                                                                                                  TRAP    C$BSEG

037214  004737  016464                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
037220  103405                                BCS     3$                      ;BR IF SOFT INIT = OK
037222  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
037224  104455                                                                                                  TRAP    C$ERDF
037226  001447                                                                                                  .WORD   807
037230  003550                                                                                                  .WORD   SFIERR
037232  011506                                                                                                  .WORD   SFIMSG
                                        3$:
037234  012704  037510                        MOV     #T8PK2,R4               ;WRITE CHARACTERISTICS PACKET
037240  004737  010152                        JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
037244  103404                                BCS     4$                      ;BR, IF COMMAND ISSUED OK
                                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
037246  104456                                                                                                  TRAP    C$ERHRD
037250  001450                                                                                                  .WORD   808
037252  004754                                                                                                  .WORD   WRTMSG
037254  011506                                                                                                  .WORD   SFIMSG
                                        4$:
037256  004737  040314                        JSR     PC,T8REST               ;SET UP PACKET FOR COMMAND
037262  012704  037440                        MOV     #T8PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
                                        5$:
037266  005037  002172                10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
037272  052714  007400                        BIS     #P.MODE,(R4)            ;NON-ZERO COMMAND MODE BITS
037276  010465  177776                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
037302  004737  017054                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
037306  103405                                BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
037310  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                                ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
037312  104455                                                                                                  TRAP    C$ERDF
037314  001451                                                                                                  .WORD   809
037316  040004                                                                                                  .WORD   T8SSR
037320  011520                                                                                                  .WORD   PKTSSR
                                        15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
037322  104406                                                                                                  TRAP    C$CLP1
                                                ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
037324  104410                                                                                                  TRAP    C$ESCAPE
037326  000074                                                                                                  .WORD   10000$-.
```

```
 6254 037330  005737  002172                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 6255 037334  001004                        BNE     22$                     ;BRANCH IF YES
 6259 037336                                ERRHRD  ERRNO,T8NINT,PKTSSR
      037336  104456                                                        TRAP    C$ERHRD
      037340  001452                                                        .WORD   810
      037342  040134                                                        .WORD   T8NINT
      037344  011520                                                        .WORD   PKTSSR
 6260 037346  016501  000000    22$:        MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 6261 037352  012702  100206                MOV     #SC!SSR!TSREJ,R2        ;EXPECTED CONTENTS OF TSSR
 6262 037356  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 6263 037362  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 6264 037364  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 6265 037370  020201    25$:                CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 6266 037372  001404                        BEQ     30$                     ;OKAY IF MATCH
 6270 037374                                ERRHRD  ERRNO,T82REJ,PKTSSR     ;COMMAND NOT REJECTED
      037374  104456                                                        TRAP    C$ERHRD
      037376  001453                                                        .WORD   811
      037400  037612                                                        .WORD   T82REJ
      037402  011520                                                        .WORD   PKTSSR
 6271 037404                        30$:
 6272 037404  004737  010354    35$:        JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
 6273 037410  103405                        BCS     59$                     ;RAM OK GO ON
 6277 037412                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
      037412  104456                                                        TRAP    C$ERHRD
      037414  001454                                                        .WORD   812
      037416  004643                                                        .WORD   PKTRAM
      037420  016200                                                        .WORD   RAMERR
 6278 037422                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      037422                                                                10000$:
      037422  104405                                                        TRAP    C$ESEG
 6279
 6280 037424                        59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      037424                                                                L10070:
      037424  104403                                                        TRAP    C$ESUB
```

```
6284 037426                              EXIT    TST                     ;ALL DONE THIS TEST
     037426  104432                                                              TRAP    C$EXIT
     037430  000770                                                              .WORD   L10066-.
6285
6286                                     ;+
6287                                     ;LOCAL STORAGE FOR THIS TEST
6288                                     ;-
6289
6291 037432                              .BLKB   10-<.-TUV2A&7>
6293 037440                              T8PACKET:                       ;COMMAND PACKET FOR TEST
6294 037440  100204                              .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
6295 037442  037450                              .WORD   T8DATA          ;ADDRESS OF CHARACTERISTICS BLOCK
6296 037444  000000                              .WORD   0
6297 037446  000010                              .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
6298
6299 037450                              T8DATA:                         ;CHARACTERISTICS DATA BLOCK
6300 037450  037462                              .WORD   T8BFR           ;ADDRESS OF MESSAGE BUFFER
6301 037452  000000                              .WORD   0
6302 037454  000016                              .WORD   14.             ;LENGTH OF MESSAGE BUFFER
6303 037456  000000  000000                      .WORD   0,0
6304
6305 037462                              T8BFR:  .BLKW   8.              ;MESSAGE BUFFER
6306
6307
6309 037502                              .BLKB   10-<.-TUV2A&7>
6311 037510                              T8PK2:                          ;COMMAND PACKET FOR TEST
6312 037510  100204                              .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
6313 037512  037520                              .WORD   T8DTA           ;ADDRESS OF CHARACTERISTICS BLOCK
6314 037514  000000                              .WORD   0
6315 037516  000010                              .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
6316
6317 037520                              T8DTA:                          ;CHARACTERISTICS DATA BLOCK
6318 037520  037532                              .WORD   T8BF2           ;ADDRESS OF MESSAGE BUFFER
6319 037522  000000                              .WORD   0
6320 037524  000016                              .WORD   14.             ;LENGTH OF MESSAGE BUFFER
6321 037526  000000  000000                      .WORD   0,0
6322
6323 037532                              T8BF2:  .BLKW   8.              ;MESSAGE BUFFER
6324
6325
6326
6327                                     ;+
6328                                     ;LOCAL TEXT MESSAGES FOR TEST
6329                                     ;-
6330
6331 037552      111     116     111     T8NBA:  .ASCIZ  'INITIALIZE Command Not Accepted'
6332 037612      111     116     111     T82REJ: .ASCIZ  'INITIALIZE Not Rejected With Non-Zero Mode Field'
6333 037673      107     105     124     T83REJ: .ASCIZ  'GET STATUS Not Accepted'
6334 037723      107     105     124     T84REJ: .ASCIZ  'GET STATUS Not Rejected With Non-Zero Mode Field'
6335 040004      103     157     156     T8SSR:  .ASCIZ  'Contents of TSSR Incorrect After INITIALIZE'
6336 040060      103     157     156     T8SR2:  .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
6337 040134      105     170     160     T8NINT: .ASCIZ  'Expected Interrupt Not Received On INITIALIZE'
6338 040212      111     156     143     T8TSBA: .ASCIZ  'Incorrect TSBA Address After INITIALIZE'
6339 040262      116     157     156     TST8ID: .ASCIZ  'Non-Tape Motion Commands'
6340                                             .EVEN
6341
```

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

040314                                  T8REST:
040314                                          SAVREG                          ;SAVE THE REGISTERS
040320  012701  037440                          MOV     #T8PACKET,R1            ;START OF THE PACKET
040324  012721  100213                          MOV     #100213,(R1)+           ;INITIALIZE WITH ACK, IE
040330  005021                                  CLR     (R1)+                   ;ADDRESS OF CHAR DATA BLOCK
040332  005021                                  CLR     (R1)+                   ;EXTENDED ADDRESS
040334  005021                                  CLR     (R1)+                   ;SIZE OF DATA BLOCK IN BYTES
040336  005021                                  CLR     (R1)+                   ;ADDRESS OF MESSAGE BUFFER
040340  005021                                  CLR     (R1)+
040342  005021                                  CLR     (R1)+                   ;LENGTH OF MESSAGE BUFFER
040344  005021                                  CLR     (R1)+
040346  005011                                  CLR     (R1)
040350  005037  037462                          CLR     T8BFR                   ;CLEAR 1ST LOC IN MESSAGE BUFFER
040354  000207                                  RTS     PC                      ;RETURN
                                        ;+
                                        ;
                                        ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
                                        ;GET STATUS COMMAND
                                        ;
                                        ;-

040356                                  T8RT2:
040356                                          SAVREG                          ;SAVE THE REGISTERS
040362  012701  037440                          MOV     #T8PACKET,R1            ;START OF THE PACKET
040366  012721  100217                          MOV     #100217,(R1)+           ;GET STATUS WITH ACK, IE
040372  005021                                  CLR     (R1)+                   ;ADDRESS OF CHAR DATA BLOCK
040374  005021                                  CLR     (R1)+                   ;EXTENDED ADDRESS
040376  005021                                  CLR     (R1)+                   ;SIZE OF DATA BLOCK IN BYTES
040400  005021                                  CLR     (R1)+                   ;ADDRESS OF MESSAGE BUFFER
040402  005021                                  CLR     (R1)+
040404  005021                                  CLR     (R1)+                   ;LENGTH OF MESSAGE BUFFER
040406  005021                                  CLR     (R1)+
040410  005011                                  CLR     (R1)
040412  005037  037462                          CLR     T8BFR                   ;CLEAR 1ST LOC IN MESSAGE BUFFER
040416  000207                                  RTS     PC                      ;RETURN
040420                                          ENDTST
040420                                                                  L10066:
040420  104401                                                                  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 ooth 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
                                ;
                                ;--

040422                                  BGNTST
040422                                                                          T9::
040422  005037  002170                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
040426  012737  005672  002146          MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
040434  005037  003100                  CLR     KTFLG           ;HOLD OFF KT11
040440  012700  042070                  MOV     #TST9ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
040444  004737  017226                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
040450  012737  000002  002164          MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
040456                          T9LOOP:                         ;LOOP ON TEST LABEL
```

```
                                        .SBTTL  TEST 9: SUBTEST 1: GET STATUS SELECTED LOCATIONS
                                ;++
                                ; TEST 9: SUBTEST 1:
                                ;
                                ; SUBTEST DESCRIPTIUN:
                                ;
                                ;       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
                                ;--
                                
040456                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
040456                                                                                  T9.1:
040456  104402                                                                          TRAP    C$BSUB
                                
                                ;Fill Memory with background pattern of 125252
040460  012700  125252                  MOV     #125252,R0              ;BACKGROUND DATA
040464  004737  020212                  JSR     PC,FILLMEM              ;FILL MEMORY WITH BACKGROUND DATA
                                
                                ;Write to TSSR to soft initialize
040470  004737  016464                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
040474  103405                          BCS     15$                     ;BR IF SOFT INIT = OK
040476                                  NEXT.ERRNO
040476  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040500                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
040500  104455                                                                          TRAP    C$ERDF
040502  001605                                                                          .WORD   901
040504  003550                                                                          .WORD   SFIERR
040506  011506                                                                          .WORD   SFIMSG
                                
                                ;Do a WRITE CHARACTERISTICS to setup a message buffer
040510                          15$:
040510  012704  041660                  MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
040514  004737  043426                  JSR     PC,T9SWRT               ;RESTORE PACKET TO STARTING VALUES
040520  005037  003102                  CLR     KTENABLE                ;TURN OFF KT-11
040524  010465  177776                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
040530  004737  017054                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
040534                                  FORCERROR       17$
                                        BCS     20$                     ;BR IF SSR SET IN CHKTSSR
```

```
6480 040552  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6481 040554                                 NEXT.ERRNO
6482 040554                         17$:    ERRDF   ERRNO,T9WRTSSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
     040554  104455                                                                TRAP    C$ERDF
     040556  001606                                                                .WORD   902
     040560  042172                                                                .WORD   T9WRTSSR
     040562  011520                                                                .WORD   PKTSSR
6483
6484                                        ;Verify a Get Status can be fetched from each address
6485                                        ;Get a valid modulo-4 test address
6486                                        ;Do a GET STATUS command from the test address
6487 040564  005037  002170                20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
6488 040570  005037  041730                        CLR     T9KT                    ;TEST ABOVE 28K SWITCH
6489 040574  012702  041734                        MOV     #T9BLK,R2               ;POINT TO TEST PATTERN TABLE
6490 040600                                 T91LOOP:
6491 040600  005037  003102                        CLR     KTENABLE                ;TURN OFF ABOVE 28K TEST FLAG
6492 040604  012201                                MOV     (R2)+,R1                ;GET TEST PATTERN ADDRESS
6493 040606  005000                                CLR     R0                      ;ASSUME NO TEST ABOVE 28K
6494 040610  005737  041730                        TST     T9KT                    ;TEST ABOVE 28K THIS TIME?
6495 040614  001407                                BEQ     25$                     ;BR IF NO
6496 040616  016200  177776                        MOV     -2(R2),R0               ;GET TEST PATTERN AGAIN
6497 040622  042700  177774                        BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
6498 040626  012737  000001  003102                MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
6499 040634  004737  042736                25$:    JSR     PC,T9CONVERT            ;CONVERT TEST PATTERN TO TEST ADDRESS
6500 040640  103034                                BCC     65$                     ;BR IF INVALID PACKET ADDRESS
6501 040642  013704  041724                        MOV     T9LOADD,R4              ;COPY CURRENT PACKET LOW ADDRESS
6502 040646  013703  041722                        MOV     T9HIADD,R3              ;CUPY CURRENT PACKET HIGH ADDRESS
6503 040652  004737  043474                        JSR     PC,T9SETGET             ;SETUP CURRENT PACKET TO GET STATUS
6504 040656  042703  177774                        BIC     #^C<A1716>,R3           ;SAVE ADDRESS BITS 17+16
6505 040662  050304                                BIS     R3,R4                   ;SETUP 18 BIT PACKET ADDRESS
6506 040664  004737  020064                        JSR     PC,KTOFF                ;TURN OFF KT-11
6507 040670  010465  177776                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6508 040674  004737  017054                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6509 040700                                        FORCERROR       32$
6510 040714  103405                                BCS     40$                     ;BR IF SSR SET IN CHKTSSR
6511 040716  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6512 040720                                        NEXT.ERRNO
6513 040720                         32$:    ERRDF   ERRNO,T9GETSSR,PKTGETS  ;DEVICE FATAL SSR FAILED TO SET
     040720  104455                                                                TRAP    C$ERDF
     040722  001607                                                                .WORD   903
     040724  042116                                                                .WORD   T9GETSSR
     040726  011550                                                                .WORD   PKTGETS
6514 040730                         40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040730  104406                                                                TRAP    C$CLP1
6515 040732                         65$:
6516 040732                                        FORCEXIT        80$
6517 040742  020227  042066                        CMP     R2,#T9TBE               ;DONE ALL TSTBLK TEST PATTERNS?
6518 040746  103002                                BHIS    70$                     ;BR IF YES
6519 040750  000137  040600                        JMP     T91LOOP         ;DO ANOTHER MODULO- 4 ADDRESS
6520 040754  005737  041730                70$:    TST     T9KT                    ;DONE ABOVE 28K TESTING TOO?
6521 040760  003012                                BGT     80$                     ;BR IF YES
6522 040762  005737  003100                        TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYSTEM?
6523 040766  001407                                BEQ     80$                     ;BR IF NO
6524 040770  012737  000001  041730                MOV     #1,T9KT         ;SET SWITCH
6525 040776  012702  041734                        MOV     #T9BLK,R2               ;RESET TEST PATTERN TABLE
6526 041002  000137  040600                        JMP     T91LOOP         ;DO ABOVE 28K TESTING
6527 041006  004737  020064                80$:    JSR     PC,KTOFF                ;TURN OFF KT11
```

```
 6528 041012                                ENDSUB                    ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
      041012                                                                    L10072:
      041012  104403                                                                TRAP    C$ESUB
 6529 041014  005737  002170                TST     FATFLG            ;ANY FATAL ERRORS ?
 6530 041020  001402                        BEQ     100$              ;BRANCH IF NOT
 6531 041022  004737  017772                JSR     PC,CKDROP         ;TRY TO DROP THE UNIT
 6532 041026                        100$:
 6533
```

```
                                        .SBTTL  TEST 9: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS
                                ;++
                                ; TEST 9: SUBTEST 2:
                                ;
                                ; SUBTEST DESCRIPTIUN:
                                ;
                                ;       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
                                ;--
041026                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                                                        T9.2:
041026  104402                                                                          TRAP    C$BSUB
                                ;Fill Memory with background pattern of 125252
041030  012700  125252                  MOV     #125252,R0              ;BACKGROUND DATA
041034  004737  020212                  JSR     PC,FILLMEM              ;FILL MEMORY WITH BACKGROUND DATA
                                ;Write to TSSR to soft initialize
041040  004737  016464                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
041044  103405                          BCS     15$                     ;BR IF SOFT INIT = OK
041046                                  NEXT.ERRNO
041046  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041050                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
041050  104455                                                                          TRAP    C$ERDF
041052  001610                                                                          .WORD   904
041054  003550                                                                          .WORD   SFIERR
041056  011506                                                                          .WORD   SFIMSG
                                ;Do a WRITE CHARACTERISTICS to setup a message buffer to compare
                                15$:
041060  012704  041660                  MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
041064  004737  043426                  JSR     PC,T9SWRT               ;SET PACKET TO WRITE CHARACTERISTICS
041070  004737  020064                  JSR     PC,KTOFF                ;TURN OFF KT-11
```

```
6586 041074  010465  177776                MOV     R4,T5DB(R5)             ;SET THE PACKET ADDRESS
6587 041100  004737  017054                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6588 041104                                FORCERROR       17$
6589 041120  103405                        BCS     20$                     ;BR IF SSR SET IN CHKTSSR
6590 041122  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6591 041124                                NEXT.ERRNO
6592 041124                        17$:    ERRDF   ERRNO,T9WRTSSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
     041124  104455                                                                 TRAP    C$ERDF
     041126  001611                                                                 .WORD   905
     041130  042172                                                                 .WORD   T9WRTSSR
     041132  011520                                                                 .WORD   PKTSSR
6593                                       ;Get a valid modulo-4 test address
6594                                       ;Set the packet message buffer to the test address
6595                                       ;Do a WRITE CHARACTERISTICS
6596 041134  005037  002170          20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
6597 041140  012703  041734                MOV     #T9BLK,R3               ;POINT TO TEST PATTERN TABLE
6598 041144                                T92LOOP:
6599 041144  012301                        MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
6600 041146  010100                        MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
6601 041150  042700  177774                BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
6602 041154  042701  000001                BIC     #1,R1                   ;ALWAYS ON A WORD BOUNDRY
6603 041160  004737  043130                JSR     PC,T9CT2                ;CONVERT TEST PATTERN TO TEST ADDRESS
6604 041164  103402                        BCS     25$                     ;BR IF VALID MESSAGE BUFFER ADDRESS
6605 041166  000137  041264                JMP     150$                    ;GET ANOTHER TEST PATTERN TO TRY
6606 041172  012704  041660          25$:    MOV     #T9PACKET,R4            ;SET THE COMMAND PACKET ADDRESS
6607 041176  004737  043426                JSR     PC,T9SWRT               ;SETUP T9PACKET TO WRITE CHAR.
6608 041202  013737  041724  041670        MOV     T9LOADD,T9DATA  ;SETUP LOW ORDER MESSAGE BUFFER ADD.
6609 041210  013737  041722  041672        MOV     T9HIADD,T9DATA+2        ;SETUP HIGH ORDER MESSAGE BUFFER ADD.
6610 041216  004737  020064                JSR     PC,KTOFF                ;TURN OFF KT-11
6611 041222  010465  177776                MOV     R4,T5DB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6612 041226  004737  017054                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6613 041232                                FORCERROR       32$
6614 041246  103405                        BCS     50$                     ;BR IF SSR SET IN CHKTSSR
6615 041250  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6616 041252                                NEXT.ERRNO
6617 041252                        32$:    ERRDF   ERRNO,T9WRTSSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
     041252  104455                                                                 TRAP    C$ERDF
     041254  001612                                                                 .WORD   906
     041256  042172                                                                 .WORD   T9WRTSSR
     041260  011520                                                                 .WORD   PKTSSR
6618 041262                        50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041262  104406                                                                 TRAP    C$CLP1
6619 041264                        150$:   FORCEXIT        160$
6620 041274  020327  042066                CMP     R3,#T9TBE               ;DONE ALL TST9BLK TEST PATTERNS?
6621 041300  103002                        BHIS    160$                    ;BR IF YES
6622 041302  000137  041144                JMP     T92LOOP         ;DO ANOTHER MODULO- 4 ADDRESS
6623 041306  004737  020064          160$:   JSR     PC,KTOFF                ;TURN OFF KT11
6624 041312                                ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     041312                                                                 L10073:
     041312  104403                                                                 TRAP    C$ESUB
6625 041314  005737  002170                TST     FATFLG                  ;ANY FATAL ERRORS ?
6626 041320  001402                        BEQ     180$                    ;BRANCH IF NOT
6627 041322  004737  017772                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
6628 041326                        180$:
```

```
                                        .SBTTL  TEST 9: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS
                                ;++
                                ; TEST 9: SUBTEST 3:
                                ;
                                ; SUBTEST DESCRIPTIUN:
                                ;
                                ;       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
                                ;--

041326                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
041326                                                                                          T9.3:
041326  104402                                                                                          TRAP    C$BSUB

                                ;Fill Memory with background pattern of 125252
041330  012700  125252                  MOV     #125252,R0              ;BACKGROUND DATA
041334  004737  020212                  JSR     PC,FILLMEM              ;FILL MEMORY WITH BACKGROUND DATA

                                ;Write to TSSR to soft initialize
041340  004737  016464                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
041344  103405                          BCS     20$                     ;BR IF SOFT INIT = OK
041346                                  NEXT.ERRNO
041346  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041350                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
041350  104455                                                                                  TRAP    C$ERDF
041352  001613                                                                                  .WORD   907
041354  003550                                                                                  .WORD   SFIERR
041356  011506                                                                                  .WORD   SFIMSG

                                ;Get a valid test address
041360  005037  002170          20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
041364  005037  041730                  CLR     T9KT                    ;TEST ABOVE 28K SWITCH
041370  012703  041734                  MOV     #T9BLK,R3               ;POINT TO TEST PATTERN TABLE
041374                                  T93LOOP:
```

```
 6681 041374  005037  003102                    CLR     KTENABLE                ;TURN OFF ABOVE 28K TEST FLAG
 6682 041400  012301                            MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
 6683 041402  010100                            MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
 6684 041404  042700  177774                    BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
 6685 041410  042701  000003                    BIC     #3,R1                   ;GET RID OF A17 AND A16
 6686 041414  005737  041730                    TST     T9KT                    ;TEST ABOVE 28K THIS TIME?
 6687 041420  001407                            BEQ     25$                     ;BR IF NO
 6688 041422  016300  177776                    MOV     -2(R3),R0               ;GET TEST PATTERN AGAIN
 6689 041426  042700  177774                    BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
 6690 041432  012737  000001  003102            MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
 6691 041440  004737  042736          25$:     JSR     PC,T9CONVERT            ;CONVERT TEST PATTERN TO TEST ADDRESS
 6692 041444  103402                            BCS     30$                     ;BR IF VALID TEST ADDRESS
 6693 041446  000137  041550                    JMP     60$                     ;GET NEXT TEST PATTERN
 6694                                 ;Set the test packet characteristics data pointer to the test address
 6695 041452  012704  041660          30$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 6696 041456  004737  043426                    JSR     PC,T9SWRT               ;RESTORE PACKET TO STARTING VALUES
 6697 041462  013764  041724  000002            MOV     T9LOADD,PKLOW(R4)       ;STORE CHAR. DATA PTR LOW ADDRESS
 6698 041470  013764  041722  000004            MOV     T9HIADD,PKHI(R4)        ;STORE CHAR. DATA PTR HIGH ADDRESS
 6699 041476  004737  043536                    JSR     PC,T9CHAR               ;STORE EXPECTED DATA IN DATA BLOCK
 6700                                 ;Do a WRITE CHARACTERISTIC command
 6701 041502  004737  020064                    JSR     PC,KTOFF                ;TURN OFF KT-11
 6702 041506  010465  177776                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 6703 041512  004737  017054                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6704 041516                                    FORCERROR       32$
 6705 041532  103405                            BCS     40$                     ;BR IF SSR SET IN CHKTSSR
 6706 041534  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6707 041536                                    NEXT.ERRNO
 6708 041536                          32$:     ERRDF   ERRNO,T9WRTSSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
      041536  104455                                                                    TRAP    C$ERDF
      041540  001614                                                                    .WORD   908
      041542  042172                                                                    .WORD   T9WRTSSR
      041544  011520                                                                    .WORD   PKTSSR
 6709 041546                          40$:     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      041546  104406                                                                    TRAP    C$CLP1
 6710 041550                          60$:
 6711 041550  020327  042066                    CMP     R3,#T9TBE               ;DONE ALL TSTBLK TEST PATTERNS?
 6712 041554  103002                            BHIS    65$                     ;BR IF YES
 6713 041556  000137  041374                    JMP     T93LOOP         ;DO ANOTHER MODULO- 4 ADDRESS
 6714 041562  005737  041730          65$:     TST     T9KT                    ;DONE ABOVE 28K TESTING TOO?
 6715 041566  003012                            BGT     70$                     ;BR IF YES
 6716 041570  005737  003100                    TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYSTEM?
 6717 041574  001407                            BEQ     70$                     ;BR IF NO
 6718 041576  012737  000001  041730            MOV     #1,T9KT         ;SET SWITCH
 6719 041604  012703  041734                    MOV     #T9BLK,R3               ;RESET TEST PATTERN TABLE
 6720 041610  000137  041374                    JMP     T93LOOP         ;DO ABOVE 28K TESTING
 6721 041614  004737  020064          70$:     JSR     PC,KTOFF                ;TURN OFF KT11
 6722 041620                                    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\
      041620                                                                    L10074:
      041620  104403                                                                    TRAP    C$ESUB
 6723 041622  005737  002170                    TST     FATFLG                  ;ANY FATAL ERRORS ?
 6724 041626  001402                            BEQ     75$                     ;BRANCH IF NOT
 6725 041630  004737  017772                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 6726 041634                          75$:
 6727 041634  004737  017174          100$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS?
 6728 041640  103002                            BCC     105$                    ;BR IF NO
 6729 041642  000137  040456                    JMP     T9LOOP                  ;LOOP UNTIL ITERATION COUNT DONE
 6730 041646                          105$:
```

```
6731 041646  004737  020064                  JSR     PC,KTOFF                ;TURN OFF MEMORY MANAGEMENT
6732 041652                                  EXIT    TST                     ;ALL DONE THIS TEST
     041652  104432                                                                  TRAP    CSEXIT
     041654  001724                                                                  .WORD   L10071-.
6733
6734
6735                                 ;+
6736                                 ;LOCAL STORAGE FOR THIS TEST
6737                                 ;-
6738
6740 041656                                  .BLKB   10-<.-TUV2A87>
6742 041660                          T9PACKET:                               ;COMMAND PACKET FOR TEST
6743 041660  100004                          .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
6744 041662  041670                          .WORD   T9DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
6745 041664  000000                          .WORD   0
6746 041666  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
6747
6748 041670                          T9DATA:                         ;CHARACTERISTICS DATA BLOCK
6749 041670  041702                          .WORD   T9BFR                   ;LOW ADDRESS OF MESSAGE BUFFER
6750 041672  000000                          .WORD   0                       ;HIGH ORDER OF MESSAGE BUFFER
6751 041674  000016                          .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
6752 041676  000000  000000                  .WORD   0,0
6753
6754 041702                          T9BFR:  .BLKW   8.                      ;MESSAGE BUFFER
6755
6756 041722  000000                  T9HIADD:        .WORD   0               ;HIGH ADDRESS
6757 041724  000000                  T9LOADD:        .WORD   0               ;LOW ADDRESS
6758 041726  000000                  T9PAR6: .WORD   0               ;ADDRESS IN PAR FORMAT
6759 041730  000000                  T9KT:           .WORD   0               ;TEST ABOVE 28K SWITCH
6760 041732  000000                  T94TST: .WORD   0               ;ADDRESS TEST BIT
6761                                 ;+
6762                                 ;
6763                                 ;TABLE OF ADDRESSES
6764                                 ;
6765                                 ;-
6766 041734  000001                  T9BLK:  .WORD   000001
6767 041736  000002                          .WORD   000002
6768 041740  000003                          .WORD   000003
6769 041742  000005                          .WORD   000005
6770 041744  000006                          .WORD   000006
6771 041746  000007                          .WORD   000007
6772 041750  000011                          .WORD   000011
6773 041752  000012                          .WORD   000012
6774 041754  000013                          .WORD   000013
6775 041756  000021                          .WORD   000021
6776 041760  000022                          .WORD   000022
6777 041762  000023                          .WORD   000023
6778 041764  000041                          .WORD   000041
6779 041766  000042                          .WORD   000042
6780 041770  000043                          .WORD   000043
6781 041772  000101                          .WORD   000101
6782 041774  000102                          .WORD   000102
6783 041776  000103                          .WORD   000103
6784 042000  000201                          .WORD   000201
6785 042002  000202                          .WORD   000202
6786 042004  000203                          .WORD   000203
6787 042006  000401                          .WORD   000401
```

```
 6788 042010  000402                          .WORD   000402
 6789 042012  000403                          .WORD   000403
 6790 042014  001001                          .WORD   001001
 6791 042016  001002                          .WORD   001002
 6792 042020  001003                          .WORD   001003
 6793 042022  002001                          .WORD   002001
 6794 042024  002002                          .WORD   002002
 6795 042026  002003                          .WORD   002003
 6796 042030  004001                          .WORD   004001
 6797 042032  004002                          .WORD   004002
 6798 042034  004003                          .WORD   004003
 6799 042036  010001                          .WORD   010001
 6800 042040  010002                          .WORD   010002
 6801 042042  010003                          .WORD   010003
 6802 042044  020001                          .WORD   020001
 6803 042046  020002                          .WORD   020002
 6804 042050  020003                          .WORD   020003
 6805 042052  040001                          .WORD   040001
 6806 042054  040002                          .WORD   040002
 6807 042056  040003                          .WORD   040003
 6808 042060  100001                          .WORD   100001
 6809 042062  100002                          .WORD   100002
 6810 042064  100003                          .WORD   100003
 6811 042066  177777                T9TBE:    .WORD           177777
 6812                                         ;+
 6813                                         ;LOCAL TEXT MESSAGES FOR TEST
 6814                                         ;-
 6815
 6816 042070     104     115     101  TST9ID: .ASCIZ  'DMA Memory Addressing'
 6817 042116     103     157     156  T9GETSSR:       .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
 6818 042172     103     157     156  T9WRTSSR:       .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
 6819 042261     115     145     163  T9MSGBUF:       .ASCIZ  'Message Buffer Contents Incorrect After WRITE CHARACTERISTICS'
 6820 042357     102     141     143  T9BKGND:        .ASCIZ  'Background Pattern Disturbed By WRITE CHARACTERISTICS'
 6821 042445     105     170     160  T9NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
 6822 042536     127     162     151  T9DPR:          .ASCIZ  'Write Characteristic data in ram does not match expected'
 6823 042627     124     123     123  T9NXM:          .ASCIZ  'TSSR NXM bit failed to set when nor existent memory address specifi
 6824                                                 .EVEN
 6825
```

```
                                        ;+
                                        ;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
                                        ;-
042736                                  T9CONVERT:
042736                                          SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
042742  005037  041724                          CLR     T9LOADD ;CLEAR LOW ADDRESS
042746  005037  041722                          CLR     T9HIADD ;CLEAR HIGH ADDRESS
042752  005037  041726                          CLR     T9PAR6          ;CLEAR PAR6 BIASED ADDRESS
042756  042701  170000                          BIC     #^C<7777>,R1    ;FORCE TO LOWER 12 BITS OF ADDRESS
042762  010005                                  MOV     R0,R5           ;SAVE HIGH ORDER ADDRESS BITS
042764  004737  020064                          JSR     PC,KTOFF        ;SHUTOFF MEMORY MANAGEMENT
042770  013702  003072                          MOV     FREE,R2         ;GET FIRST FREE ADDRESS
042774  062702  000020                          ADD     #16.,R2         ;IN CASE TEST PATTERN=0
043000  060102                                  ADD     R1,R2           ;ADD IN TEST PATTERN
043002  042702  000003                          BIC     #3,R2           ;MAKE IT MODULO-4
043006  013703  003076          25$:            MOV     FREEHI,R3       ;GET LAST FREE ADDRESS
043012  162703  000020                          SUB     #16.,R3         ;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
043016  010237  041724                          MOV     R2,T9LOADD      ;SAVE POSSIBLE LOW ADDRESS
043022  010237  041726                          MOV     R2,T9PAR6       ;SAVE IT IN PAR6 BIASED TOO
043026  020203                                  CMP     R2,R3           ;IS THIS ADDRESS ABOVE FREE SPACE?
043030  101007                                  BHI     35$             ;BR IF YES
043032  020237  003072                          CMP     R2,FREE         ;IS IT IN FREE SPACE?
043036  103007                                  BHIS    50$             ;BR IF YES- ITS GOOD
043040  005737  003102                          TST     KTENABLE        ;TESTING ABOVE 28K?
043044  001004                                  BNE     50$             ;BR IF YES
043046  000424                                  BR      90$             ;BR IF NOT IN FREE SPACE
043050  162702  000020          35$:            SUB     #16.,R2         ;FORCE FIT THE TEST PATTERN
043054  000754                                  BR      25$             ;TRY THIS TEST PATTERN ADDRESS
043056  005737  003102          50$:            TST     KTENABLE        ;TESTING ABOVE 28K?
043062  001420                                  BEQ     100$            ;BR IF NO
043064  005737  003100                          TST     KTFLG           ;ANY MEMORY ABOVE 28K?
043070  001413                                  BEQ     90$             ;BR IF NO
043072  004737  020046                          JSR     PC,KTON         ;TURN ON MEMORY MANAGEMENT
043076  010500                                  MOV     R5,R0           ;GET HIGH ORDER ADDRESS
043100  010037  041722                          MOV     R0,T9HIADD      ;SAVE POSSIBLE HIGH ADDRESS
043104  010201                                  MOV     R2,R1           ;GET COMPUTED LOW ORDER ADDRESS
043106  004737  020106                          JSR     PC,SETMAP       ;RETURN PAR6 BIASED ADDRESS IN R0
043112  010037  041726                          MOV     R0,T9PAR6       ;COPY PAR6 BIASED ADDRESS
043116  103403                                  BCS     105$            ;BR IF VALID ADDRESS
043120  000241          90$:            CLC                             ;CLR C BIT FOR FAILURE
043122  000401                                  BR      105$            ;
043124  000261          100$:           SEC                             ;SET SUCCESS
043126  000207          105$:           RTS     PC                      ;RETURN
```

```
                                        ;+
                                        ;ONLY FOR MESSAGE BUFFER ADDRESSES
                                        ;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
                                        ;-
043130                                  T9CT2:  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
043134  005037  041724                          CLR     T9LOADD ;CLEAR LOW ADDRESS
043140  005037  041722                          CLR     T9HIADD ;CLEAR HIGH ADDRESS
043144  005037  041726                          CLR     T9PAR6          ;CLEAR PAR6 BIASED ADDRESS
043150  042701  170000                          BIC     #^C<7777>,R1    ;FORCE TO LOWER 12 BITS OF ADDRESS
043154  010005                                  MOV     R0,R5           ;SAVE HIGH ORDER ADDRESS BITS
043156  004737  020064                          JSR     PC,KTOFF        ;SHUTOFF MEMORY MANAGEMENT
043162  013702  003072                          MOV     FREE,R2         ;GET FIRST FREE ADDRESS
043166  062702  000020                          ADD     #16.,R2         ;IN CASE TEST PATTERN=0
043172  060102                                  ADD     R1,R2           ;ADD IN TEST PATTERN
043174  013703  003076                25$:     MOV     FREEHI,R3       ;GET LAST FREE ADDRESS
043200  162703  000020                          SUB     #16.,R3         ;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
043204  010237  041724                          MOV     R2,T9LOADD      ;SAVE POSSIBLE LOW ADDRESS
043210  010237  041726                          MOV     R2,T9PAR6       ;SAVE IT IN PAR6 BIASED TOO
043214  020203                                  CMP     R2,R3           ;IS THIS ADDRESS ABOVE FREE SPACE?
043216  101007                                  BHI     35$             ;BR IF YES
043220  020237  003072                          CMP     R2,FREE         ;IS IT IN FREE SPACE?
043224  103007                                  BHIS    50$             ;BR IF YES- ITS GOOD
043226  005737  003102                          TST     KTENABLE        ;TESTING ABOVE 28K?
043232  001004                                  BNE     50$             ;BR IF YES
043234  000424                                  BR      90$             ;BR IF NOT IN FREE SPACE
043236  162702  000020                35$:     SUB     #16.,R2         ;FORCE FIT THE TEST PATTERN
043242  000754                                  BR      25$             ;TRY THIS TEST PATTERN ADDRESS
043244  005737  003102                50$:     TST     KTENABLE        ;TESTING ABOVE 28K?
043250  001420                                  BEQ     100$            ;BR IF NO
043252  005737  003100                          TST     KTFLG           ;ANY MEMORY ABOVE 28K?
043256  001413                                  BEQ     90$             ;BR IF NO
043260  004737  020046                          JSR     PC,KTON         ;TURN ON MEMORY MANAGEMENT
043264  010500                                  MOV     R5,R0           ;GET HIGH ORDER ADDRESS
043266  010037  041722                          MOV     R0,T9HIADD      ;SAVE POSSIBLE HIGH ADDRESS
043272  010201                                  MOV     R2,R1           ;GET COMPUTED LOW ORDER ADDRESS
043274  004737  020106                          JSR     PC,SETMAP       ;RETURN PAR6 BIASED ADDRESS IN R0
043300  010037  041726                          MOV     R0,T9PAR6       ;COPY PAR6 BIASED ADDRESS
043304  103403                                  BCS     105$            ;BR IF VALID ADDRESS
043306  000241                90$:     CLC                             ;CLR C BIT FOR FAILURE
043310  000401                                  BR      105$            ;
043312  000261                100$:    SEC                             ;SET SUCCESS
043314  000207                105$:    RTS     PC              ;RETURN
```

```
                                        ;+
                                        ;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
                                        ;-
043316                                  T9CKRAM::
043316                                          SAVREG                          ;SAVE THE GENERAL REGISTERS
043322  012701  002206                          MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
043326  012702  000020                          MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
043332  005003                                  CLR     R3                      ;CLEAR THE ERROR FLAG
043334  004737  017054                          JSR     PC,CHKTSSR              ;WAIT FOR SSR
043340  112765  000000  177776                  MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
043346  004737  017054                  10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
043352  010265  177776                          MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
043356  004737  017054                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
043362  116511  177776                          MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
043366  122124                                  CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
043370  001401                                  BEQ     20$                     ;BRANCH IF OK
043372  005203                                  INC     R3                      ;SET ERROR FLAG
043374  005202                          20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
043376  020227  000022                          CMP     R2,#RMPKTBEG+2          ;DONE 2 BYTES?
043402  002761                                  BLT     10$                     ;BR IF NO
043404  005703                                  TST     R3                      ;WAS AN ERROR FOUND ?
043406  001402                                  BEQ     30$                     ;BRANCH IF NOT
043410  000241                                  CLC                             ;CLEAR CARRY TO SHOW ERROR
043412  000401                                  BR      50$                     ;AND EXIT
043414  000261                          30$:    SEC                             ;SHOW GOOD COMPARE
043416  012737  000002  002246          50$:    MOV     #2,RAMSIZ               ;SETUP RAMSIZ
043424  000207                                  RTS     PC                      ;RETURN
                                        ;+
                                        ;
                                        ;ROUTINE TO SETUP PACKET TO WRITE CHARACTERISTICS
                                        ;-
043426                                  T9SWRT:
043426                                          SAVREG                          ;SAVE THE REGISTERS
043432  012701  041660                          MOV     #T9PACKET,R1            ;START OF THE PACKET
043436  012721  100004                          MOV     #100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
043442  012721  041670                          MOV     #T9DATA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
043446  005021                                  CLR     (R1)+                   ;EXTENDED ADDRESS
043450  012721  000010                          MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
043454  012721  041702                          MOV     #T9BFR,(R1)+            ;ADDRESS OF MESSAGE BUFFER
043460  005021                                  CLR     (R1)+
043462  012721  000016                          MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
043466  005021                                  CLR     (R1)+
043470  005011                                  CLR     (R1)
                                                RTS     PC                      ;RETURN
```

```
                                        ;+
                                        ;ROUTINE TO SETUP A GET STATUS COMMAND PACKET AT CURRENT PACKET ADDRESS
                                        ;       R3      HIGH ORDER PACKET ADDRESS
                                        ;       R4      LOW ORDER PACKET ADDRESS
                                        ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG CLEAR
                                        ;-
043474                                  T9SETGET:
043474                                          SAVREG                  ;SAVE THE REGISTERS
043500  010401                                  MOV     R4,R1           ;GET LOW ORDER ADDRESS
043502  005737  003102                          TST     KTENABLE        ;TESTING ABOVE 28K?
043506  001404                                  BEQ     10$             ;BR IF NO
043510  010300                                  MOV     R3,R0           ;GET HIGH ORDER ADDRESS
043512  004737  020106                          JSR     PC,SETMAP       ;RETURN ADDRESS BIASED TO PAR6 IN R0
043516  010001                                  MOV     R0,R1           ;GET ADDRESS
043520  012700  000017          10$:            MOV     #P.GETSTATUS,R0 ;GET STATUS COMMAND CODE NO IE
043524  052700  100000                          BIS     #P.ACK,R0       ;SET ACK
043530  010021                                  MOV     R0,(R1)+        ;STORE GET STATUS IN PACKET
043532  005021                                  CLR     (R1)+           ;CLEAR UNUSED WORD
043534  000207                                  RTS     PC              ;RETURN
                                        ;+
                                        ;ROUTINE TO SETUP A CHARACTERISTIC DATA BLOCK AT A TEST ADDRESS
                                        ;-
043536                                  T9CHAR:
043536                                          SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
043542  012700  041670                          MOV     #T9DATA,R0      ;GET T9PACKET DATA POINTER
043546  013701  041724                          MOV     T9LOADD,R1      ;ASSUME NOT ABOVE 28K
043552  005737  003102                          TST     KTENABLE        ;TESTING ABOVE 28K?
043556  001402                                  BEQ     10$             ;BR IF NO
043560  013701  041726                          MOV     T9PAR6,R1       ;SET TEST ADDRESS ABOVE 28K
043564  012021                  10$:            MOV     (R0)+,(R1)+     ;STORE DATA WORD 1
043566  012021                                  MOV     (R0)+,(R1)+     ;STORE DATA WORD 2
043570  012021                                  MOV     (R0)+,(R1)+     ;STORE DATA WORD 3
043572  012021                                  MOV     (R0)+,(R1)+     ;STORE DATA WORD 4
043574  012021                                  MOV     (R0)+,(R1)+     ;STORE DATA WORD 5
043576  000207                                  RTS     PC              ;RETURN
043600                                          ENDTST
043600                                                          L10071:
043600  104401                                                          TRAP    CSETST
```

```
                                        .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::
043602 005037 002170                    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
043606 012737 005672 002146             MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
043614 005037 003100                    CLR     KTFLG           ;HOLD OFF KT11
043620 012700 044252                    MOV     #TST10ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
043624 004737 017226                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
043630 012737 000002 002164             MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
043636                          T10LOOP:
043636 004737 044526                    JSR     PC,T10REST              ;SET PACKET TO START-UP VALUES

043642 012703 002732                    MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
043646 012704 044210                    MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
043652 012764 000010 000006             MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
043660                          5$:
043660 004737 016464                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
043664 103405                           BCS     10$                     ;BR IF SOFT INIT OKAY
043666 010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043670                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
043670 104455                                                                   TRAP    C$ERDF
043672 001750                                                                   .WORD   1000
043674 003550                                                                   .WORD   SFIERR
043676 011506                                                                   .WORD   SFIMSG

                                ;Do WRITE CHARACTERISTICS command.
043700 005037 002170            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
043704 010465 177776                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
043710 004737 017054                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
043714                                  FORCERROR       12$                     ;@@FORCE ERROR IF FCRCER=1
043730 103407                           BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
043732 010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043734                                  NEXT.ERRNO
043734                          12$:    ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
043734 104455                                                                   TRAP    C$ERDF
043736 001751                                                                   .WORD   1001
043740 044637                                                                   .WORD   T10SSR
```

```
     043742  011520                                                                       .WORD    PKTSSR
     043744  004737  017720            JSR     PC,FATCHK                    ;INC AND CHECK FOR MORE THAN 25 ERRORS
     043750                    15$:    CKLOOP                               ;LOOP ON ERROR, IF FLAG SET
     043750  104406                                                                       TRAP     C$CLP1
     043752  016501  000000            MOV     TSSR(R5),R1                  ;GET THE CONTENTS OF TSSR
     043756  012702  000^00            MOV     #SSR,R2                      ;EXPECTED CONTENTS OF TSSR
     043762  032701  00( 30            BIT     #OFL,R1                      ;IS OFF-LINE BIT SET ?
     043766  001402                    BEQ     25$                          ;BRANCH IF NOT OFF-LINE
     043770  052702  000100            BIS     #OFL,R2                      ;SET OFF-LINE IN EXPECTED DATA

                               ;If the NBA bit in the TSSR register is NOT=0 then Print Error.
     043774                    25$:
     043774                            FORCERROR       27$                         ;ddd
     044010  020201                    CMP     R2,R1                        ;DOES EXPECTED MATCH RECEIVED ?
     044012  001404                    BEQ     30$                          ;OKAY IF MATCH
     044014                            NEXT.ERRNO
     044014                    27$:    ERRHRD  ERRNO,T10NBA,PKTSSR          ;NBA NOT ZERO
     044014  104456                                                                       TRAP     C$ERHRD
     044016  001752                                                                       .WORD    1002
     044020  044364                                                                       .WORD    T10NBA
     044022  011520                                                                       .WORD    PKTSSR
     044024                    30$:    CKLOOP                               ;LOOP ON ERROR ?
     044024  104406                                                                       TRAP     C$CLP1

                               ;Write to TSSR register to soft initialize the controller
     044026                    40$:
     044026  004737  016464            JSR     PC,SOFINIT                   ;WRITE TO TSSR TO SOFT INITIALIZE
     044032                            FORCERROR       42$                         ;ddd
     044046  103405                    BCS     50$                          ;BR IF SOFT INIT OKAY
     044050  010001                    MOV     R0,R1                        ;SAVE CONTENTS OF TSSR
     044052                            NEXT.ERRNO
     044052                    42$:    ERRDF   ERRNO,SFIERR,SFIMSG          ;DEVICE FATAL DURING INIT
     044052  104455                                                                       TRAP     C$ERDF
     044054  001753                                                                       .WORD    1003
     044056  003550                                                                       .WORD    SFIERR
     044060  011506                                                                       .WORD    SFIMSG

                               ;If controller RAM 310-377 NOT=0 then Print Error
     044062  012704  000310    50$:    MOV     #310,R4                      ;START WITH LOC 310
     044066  005002                    CLR     R2                           ;MEMORY EXPECTED SHOULD BE 000000
     044070  004737  017054            JSR     PC,CHKTSSR                   ;WAIT FOR SSR READY
     044074  110465  177777    60$:    MOVB    R4,TSDBH(R5)                 ;SELECT RAM ADDRESS
     044100  116501  177776            MOVB    TSBAL(R5),R1                 ;READ LOC CONTENTS
     044104                            FORCERROR       62$,NOTSSR                  ;ddd
     044114  120102                    CMPB    R1,R2                        ;CHECK MEMORY FOR 000000
     044116  001406                    BEQ     70$                          ;BRANCH IF DATA OKAY
     044120                            NEXT.ERRNO
     044120                    62$:    ERRDF   ERRNO,T10MEM,RAMEXP          ;MEMORY NOT ZERO AFTER INIT.
     044120  104455                                                                       TRAP     C$ERDF
     044122  001754                                                                       .WORD    1004
     044124  044325                                                                       .WORD    T10MEM
     044126  016220                                                                       .WORD    RAMEXP
     044130  004737  017720            JSR     PC,FATCHK                    ;INC AND CHECK FOR MORE THAN 25 ERRORS
     044134                    70$:    CKLOOP
     044134  104406                                                                       TRAP     C$CLP1
     044136                            ESCAPE  TST                          ;EXIT ON FATAL ERROR
     044136  104410                                                                       TRAP     C$ESCAPE
```

```
                044140  000434                                                          .WORD   L10075-.
 7128
 7129 044142  005204                    82$:    INC     R4                      ;LOOK AT NEXT RAM LOC.
 7130 044144  020427  000377                    CMP     R4,#377                 ;AT TOP OF RAM ADDRESS SPACE
 7131 044150  001351                            BNE     60$                     ;BRANCH TILL ALL MEMORY TESTED
 7132
 7133
 7134 044152  005737  002170                    TST     FATFLG                  ;ANY FATAL ERRORS ?
 7135 044156  001402                            BEQ     160$                    ;BRANCH IF NOT
 7136 044160  004737  017772                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 7137 044164  004737  017174            160$:   JSR     PC,TSTLOOP              ;DONE ALL ITERATIONS?
 7138 044170  103002                            BCC     165$                    ;BR IF YES
 7139 044172  000137  043636                    JMP     T10LOOP                 ;LOOP UNTIL ITERATION COUNT DONE
 7140 044176                            165$:
 7141 044176                                    EXIT    TST
                044176  104432                                                          TRAP    CSEXIT
                044200  000374                                                          .WORD   L10075-.
 7142                                   ;+
 7143                                   ;LOCAL STORAGE FOR THIS TEST
 7144                                   ;-
 7146 044202                                    .BLKB   10-<.-TUV2A87>
 7148 044210                            T10PACKET:
 7149 044210  100004                            .WORD   100004                  ;COMMAND PACKET FOR TEST
 7150 044212  044220                            .WORD   T10DATA                 ;WRITE CHARACTERISTICS COMMAND, WITH ACK
 7151 044214  000000                            .WORD   0                       ;ADDRESS OF CHARACTERISTICS BLOCK
 7152 044216  000010                            .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
 7153
 7154 044220                            T10DATA:                                        ;CHARACTERISTICS DATA BLOCK
 7155 044220  044232                            .WORD   T10BFR                  ;ADDRESS OF MESSAGE BUFFER
 7156 044222  000000                            .WORD   0
 7157 044224  000016                            .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
 7158 044226  000000  000000                    .WORD   0,0
 7159
 7160 044232                            T10BFR: .BLKW   8.                      ;MESSAGE BUFFER
 7161                                   ;LOCAL TEXT MESSAGES FOR TEST
 7162                                   ;-
 7163
 7164 044252      111     156     151   TST10ID:        .ASCIZ  'Initialization After WRITE CHARACTERISTICS'
 7165 044325      111     156     143   T10MER: .ASCIZ  'Incorrect RAM Data After Init'
 7166                                           .EVEN
 7167 044364      127     122     111   T10NBA: .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
 7168 044437      103     157     156   T10SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
 7169
```

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

                                        .EVEN

                                        T10REST:
                                                SAVREG                          ;SAVE THE REGISTERS
                                                MOV     #T10PACKET,R1           ;START OF THE PACKET
                                                MOV     #100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
                                                MOV     #T10DATA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
                                                CLR     (R1)+                   ;EXTENDED ADDRESS
                                                MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
                                                MOV     #T10BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
                                                CLR     (R1)+
                                                MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
                                                CLR     (R1)+
                                                CLR     (R1)
                                                RTS     PC                      ;RETURN
                                                ENDTST
                                                                                                L10075:
                                                                                                        TRAP    C$ETST
```

```
7195
7196                                            .SBTTL  TEST 11: BASIC WRITE SUBSYSTEM MEMORY COMMAND
7197                                    ;+
7198                                    ;
7199                                    ;THIS TEST VERIFIES THAT THE WRITE SUBSYSTEM MEMORY COMMAND WITH A
7200                                    ;BSEL0 SELECT CODE OF 0 (NO-OP) EXECUTES CORRECTLY. IT ALSO
7201                                    ;VERIFIES THAT A WRITE SUBSYSTEM MEMORY COMMAND WITH A NON-ZERO
7202                                    ;MODE FIELD IS REJECTED. THE TEST FURTHER VERIFIES MICROPROGRAM
7203                                    ;COMMAND DECODING AND HANDLING SEQUENCES.
7204                                    ;
7205                                    ;-
7206 044576                                     BGNTST
      044576                                                                    T11::
7207 044576  005037  002170                      CLR     FATFLG                 ;CLEAR FATAL ERROR FLAG
7208 044602  012737  005672  002146              MOV     #EPRT1,EPRTSW          ;SET UP ERROR MESSAGE SWITCH
7209 044610  005037  003100                      CLR     KTFLG          ;HOLD OFF KT11
7214 044614  012700  045660                      MOV     #TST11ID,R0            ;ASCII MESSAGE TO IDENTIFY TEST
7215 044620  004737  017226                      JSR     PC,TSTSETUP            ;DO INITIAL TEST SETUP
7216 044624  012737  000002  002164              MOV     #2.,LOOPCNT            ;PERFORM 2 ITERATIONS
7217 044632                             T11LOOP:
7218 044632                                     BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      044632                                                                    T11.1:
      044632  104402                                                                   TRAP    C$BSUB
7219 044634  004737  045726                      JSR     PC,T11REST             ;SET PACKET TO INITIAL VALUES
7220 044640  004737  045764                      JSR     PC,T11RST              ;SET PACKET TO INITIAL VALUES
7221 044644                                     SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      044644  012700  000000                                                            MOV     #PRI00,R0
      044650  104441                                                                    TRAP    C$SPRI
7222 044652                             5$:
7223 044652                                     BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      044652  104404                                                                    TRAP    C$BSEG
7224 044654  004737  016464                      JSR     PC,SOFINIT             ;DO SOFT INIT OF CONTROLLER
7225 044660  103405                              BCS     10$                    ;BR IF SOFT INIT = OK
7229 044662  010001                              MOV     R0,R1                  ;SAVE CONTENTS OF TSSR
7230 044664                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      044664  104455                                                                    TRAP    C$ERDF
      044666  002115                                                                    .WORD   1101
      044670  003550                                                                    .WORD   SFIERR
      044672  011506                                                                    .WORD   SFIMSG
7231 044674                             10$:
7232 044674  012704  045470                      MOV     #T11PK2,R4             ;SUBROUTINE NEEDS PACKET ADDRESS
7233 044700  004737  010152                      JSR     PC,WRTCHR              ;ISSUE WRITE CHARACTERISTICS
7234 044704  103405                              BCS     11$                    ;BR, IF COMMAND ISSUED OK
7238 044706  010001                              MOV     R0,R1                  ;SAVE CONTENTS OF TSSR
7239 044710                                     ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      044710  104456                                                                    TRAP    C$ERHRD
      044712  002116                                                                    .WORD   1102
      044714  004754                                                                    .WORD   WRTMSG
      044716  011506                                                                    .WORD   SFIMSG
7240 044720                             11$:
7241 044720  005037  002170                      CLR     FATFLG                 ;CLEAR FATAL ERROR FLAG
7242 044724  005037  002172                      CLR     INTRECV                ;CLEAR INTERRUPT RECEIVED FLAG
7243 044730  012704  045050                      MOV     #T11PACKET,R4          ;SET UP NEW WRT. SUBSYS MEM. COMMAND
7244 044734  010465  177776                      MOV     R4,TSDB(R5)            ;SET THE PACKET ADDRESS
7245 044740  004737  017054                      JSR     PC,CHKTSSR             ;WAIT FOR SSR TO SET
7246 044744  005737  002172                      TST     INTRECV                ;DID AN INTERRUPT OCCUR ?
7247 044750  001004                              BNE     22$                    ;BRANCH IF YES
```

```
7251 044752                                    ERRHRD  ERRNO,T11NINT,PKTSSR
     044752  104456                                                    TRAP    C$ERHRD
     044754  002117                                                    .WORD   1103
     044756  045566                                                    .WORD   T11NINT
     044760  011520                                                    .WORD   PKTSSR
7252 044762  016501  000000          22$:      MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
7253 044766  012702  000200                    MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
7254 044772  032701  000100                    BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
7255 044776  001402                            BEQ     25$             ;BRANCH IF NOT OFF-LINE
7256 045000  052702  000100                    BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
7257 045004  020201                  25$:      CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
7258 045006  001404                            BEQ     30$             ;OKAY IF MATCH
7262 045010                                    ERRHRD  ERRNO,T11NBA,PKTSSR     ;NBA NOT ZERO
     045010  104456                                                    TRAP    C$ERHRD
     045012  002120                                                    .WORD   1104
     045014  045512                                                    .WORD   T11NBA
     045016  011520                                                    .WORD   PKTSSR
7263 045020                          30$:
7264 045020                                    ENDSEG                  ;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
     045020                                                        10000$:
     045020  104405                                                    TRAP    C$ESEG
7265
7266
7267 045022                          59$:      ENDSUB                  ;\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\
     045022                                                        L10077:
     045022  104403                                                    TRAP    C$ESUB
7268
7269 045024  005737  002170                    TST     FATFLG          ;ANY FATAL ERRORS ?
7270 045030  001402                            BEQ     60$             ;BRANCH IF NOT
7271 045032  004737  017772                    JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
7272 045036                          60$:
7273 045036                                    EXIT    TST             ;ALL DONE THIS TEST
     045036  104432                                                    TRAP    C$EXIT
     045040  000776                                                    .WORD   L10076-.
7274
7275                                 ;+
7276                                 ;LOCAL STORAGE FOR THIS TEST
7277                                 ;-
7278
7280 045042                                    .BLKB   10-<.-TUV2A&7>
7282 045050                          T11PACKET:                        ;COMMAND PACKET FOR TEST
7283 045050  100206                            .WORD   100206          ;WRITE SUBSYSTEM MEM. COMMAND, WITH IE, ACK
7284 045052  045060                            .WORD   T11DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
7285 045054  000000                            .WORD   0
7286 045056  000006                            .WORD   6.              ;STARTING VALUE OF BLOCK SIZE
7287 045060                          T11DATA:                          ;CHARACTERISTICS DATA BLOCK
7288 045060     000                  T11BS0: .BYTE   0                 ;BSEL0 BYTE
7289 045061     000                  T11BS1: .BYTE   0                 ;BSEL1 BYTE
7290 045062  000000                T11BS2: .WORD   0                 ;BSEL1 WORD
7291 045064  000000                        .WORD   0                 ;DATA
7292 045066                          T11BFR: .BLKW   128.              ;MESSAGE BUFFER
7293
7294
7296 045466                                    .BLKB   10-<.-TUV2A&7>
7298 045470                          T11PK2:                           ;COMMAND PACKET FOR TEST
7299 045470  100204                            .WORD   100204          ;WRITE CHARA. MEM. CMND., WITH IE, ACK
7300 045472  045500                            .WORD   T11DTA          ;ADDRESS OF SELECT DATA BLOCK
```

```
7301 045474  000000                          .WORD   0
7302 045476  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
7303
7304
7305 045500                          T11DTA:                                 ;SELECT DATA BLOCK
7306 045500  045066                          .WORD   T11BFR                  ;ADDRESS OF MESSAGE BUFFER
7307 045502  000000                          .WORD   0
7308 045504  000400                          .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
7309 045506  000000  000000                  .WORD   0,0
7310
7311
7312                                 ;+
7313                                 ;LOCAL TEXT MESSAGES FOR TEST
7314                                 ;-
7315
7316 045512     127     122     111  T11NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
7317 045566     105     170     160  T11NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE SUBSYSTEM MEMORY'
7318 045660     102     141     163  TST11ID:        .ASCIZ  'Basic WRITE SUBSYSTEM MEMORY Command'
7319                                         .EVEN
7320
```

```
7322
7323                                    ;+
7324                                    ;
7325                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
7326                                    ;WRITE SUBSYSTEM MEMORY COMMAND
7327                                    ;
7328                                    ;-
7329
7330 045726                             T11REST:
7331 045726                                     SAVREG                          ;SAVE THE REGISTERS
7332 045732  012701  045050                      MOV     #T11PACKET,R1           ;START OF THE PACKET
7333 045736  012721  100206                      MOV     #100206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
7334 045742  012721  045060                      MOV     #T11DATA,(R1)+          ;ADDRESS OF DATA BLOCK
7335 045746  005021                              CLR     (R1)+                   ;EXTENDED ADDRESS
7336 045750  012721  000006                      MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
7337 045754  005021                              CLR     (R1)+                   ;CLEAR BSEL0 AND BSEL1
7338 045756  005021                              CLR     (R1)+                   ;CLEAR SEL2
7339 045760  005011                              CLR     (R1)                    ;CLEAR DATA AREA
7340 045762  000207                              RTS     PC                      ;RETURN
7341
7342
7343 045764                             T11RST:
7344 045764                                     SAVREG                          ;SAVE THE REGISTERS
7345 045770  012701  045470                      MOV     #T11PK2,R1              ;START OF THE PACKET
7346 045774  012721  100204                      MOV     #100204,(R1)+           ;WRITE CHARA. WITH ACK, IE
7347 046000  012721  045500                      MOV     #T11DTA,(R1)+           ;ADDRESS OF CHARAISTICS DATA BLOCK
7348 046004  005021                              CLR     (R1)+                   ;EXTENDED ADDRESS
7349 046006  012721  000010                      MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
7350 046012  012721  045066                      MOV     #T11BFR,(R1)+           ;MESSAGE BUFFER ADDRESS
7351 046016  005021                              CLR     (R1)+
7352 046020  012721  000400                      MOV     #256.,(R1)+             ;LENGTH OF MESSAGE BUFFER
7353 046024  005021                              CLR     (R1)+
7354 046026  005011                              CLR     (R1)
7355 046030  005057  045066                      CLR     T11BFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
7356 046034  000207                              RTS     PC                      ;RETURN
7357 046036                             ENDTST
     046036                                                     L10076:
     046036  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.
                                ;--
052636                                  BGNHRD
052636  000015                          .WORD L10100-L$HARD/2
052640                          L$HARD::

052640                                  GPRMA   HPM1,0,0,160000,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
052640  000031                          .WORD   T$CODE
052642  052672                          .WORD   HPM1
052644  160000                          .WORD   T$LOLIM
052646  177776                          .WORD   T$HILIM
052650                                  GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
052650  001031                          .WORD   T$CODE
052652  052721                          .WORD   HPM2
052654  000000                          .WORD   T$LOLIM
052656  000776                          .WORD   T$HILIM
052660                                  GPRMD   HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
052660  002032                          .WORD   T$CODE
052662  052745                          .WORD   HPM3
052664  000340                          .WORD   340
052666  000000                          .WORD   T$LOLIM
052670  000007                          .WORD   T$HILIM
052672                                  ENDHRD
                                        .EVEN

052672                          L10100:
052672  104     105     126     HPM1:   .ASCIZ  'DEVICE ADDRESS (TSSR) '
052721  111     116     124     HPM2:   .ASCIZ  'INTERRUPT VECTOR    '
052745  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.
                                ;--
                                        BGNSFT
052776                          .WORD L10101-L$SOFT/2
053000                  L$SOFT::
                                        GPRML   SPM1,0,-1,YES                   ;GET RAM DUMP TEST FLAG
053000  000130                  .WORD   T$CODE
053002  053014                  .WORD   SPM1
053004  177777                  .WORD   -1
                                        GPRML   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
053006  001130                  .WORD   T$CODE
053010  053053                  .WORD   SPM4
053012  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
                                        ENDSFT
                                        .EVEN
053014                          L10101:

053014  105 116 101             SPM1:   .ASCIZ  'ENABLE M7454 RAM DUMP ON ERROR'
053053  111 116 110             SPM4:   .ASCIZ  'INHIBIT ITERATIONS     '
053103  120 105 122             SPM6:   .ASCIZ  'PER TEST ERROR LIMIT   '
053133  120 105 122             SPM7:   .ASCIZ  'PER UNIT ERROR LIMIT   '
                                        .EVEN
                                        .SBTTL  PATCH AREA
                                ;+
                                ;DISPATCH TABLE
                                ;
                                ; *** MOVE TO FRONT OF PROGRAM FOR RELEASE ***
                                ;-

                                        DISPATCH        TESTNO
053164  000013                  .WORD   11
053166                  L$DISPATCH::
053166  023464                  .WORD   T1
053170  023722                  .WORD   T2
053172  024674                  .WORD   T3
053174  026154                  .WORD   T4
053176  030206                  .WORD   T5
053200  031416                  .WORD   T6
053202  033472                  .WORD   T7
053204  036704                  .WORD   T8
053206  040422                  .WORD   T9
053210  043602                  .WORD   T10
053212  044576                  .WORD   T11

                                ;
                                ; FINALLY A GENEROUS PATCH AREA.
```

```
 8502                                   ;
 8503                                   ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
 8504                                   ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
 8505                                   ;
 8506
 8507 053214                            PATCH::
 8508                                   ;       .IF     NZ,.&377
 8509                                   ;       .=.!377+1
 8510                                   ;       .ENDC
 8511 053214                                    LASTAD          ;SET LAST USED ADDRESS.
                                                .EVEN
      053214  053232                            .WORD T$FREE
      053216  000005                            .WORD T$SIZE
      053220                            L$LAST::
 8512                                           .SBTTL  HARD CODED P-TABLE
 8513                                   ;++
 8514                                   ;       DIAGNOSTIC IS PRE-PARAMETERIZED PER THIS TABLE
 8515                                   ;--
 8516 053220                                    BGNSETUP        1
 8517 053220                                    BGNPTAB
      053220  000000                            .WORD   0
      053222  000003                            .WORD   L10104-./2-1
      053224                            L10102:
 8518 053224  172522                            .WORD           172522
 8519 053226  000224                            .WORD           224
 8520 053230  000240                            .WORD           PRI05
 8521 053232                                    ENDPTAB
      053232                            L10104:
 8522 053232                                    ENDSETUP
 8523
 8524         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  006151        C$BRK = 000022      DEVDRO  023414        FERCM   011374        HOE   = 100000 G
ASSEMB= 000010        C$BSEG= 000004      DEVNRD  023333        FIFEXP  011642 G      HPM1    052672
A1716 = 000003        C$BSUB= 000002      DEVNXR  023251        FIF1MS  011714        HPM2    052721
BADDAT  003110 G      C$CEFG= 000045      DEVONL  023201        FIF2MS  011763        HPM3    052745
BADSSR  016370 G      C$CLCK= 000062      DEVSUM  023144        FILLME  020212        IBE   = 010000 G
BAR   = 174402        C$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$CVEC= 000036      DLCYL = 000177        FREEHI  003076        INCERR  017562
BIT00 = 000001 G      C$DCLM= 000044      DLDNER= 100200        FRESIZ  003074 G      INTCPC  016640
BIT01 = 000002 G      C$DODU= 000051      DLERR = 177730        FUSI    004015        INTFLA  016635
BIT02 = 000004 G      C$DRPT= 000024      DLGETS= 000004        F$AU  = 000015        INTMAS  016634
BIT03 = 000010 G      C$DU  = 000053      DLRDHD= 000010        F$AUTO= 000020        INTR    016706 G
BIT04 = 000020 G      C$EDIT= 000003      DLRDNH= 000016        F$BGN = 000040        INTREC  002172 G
BIT05 = 000040 G      C$ERDF= 000055      DLSR  = 000013        F$CLEA= 000007        INTVEC  016636
BIT06 = 000100 G      C$ERHR= 000056      DLUN  = 000006        F$DU  = 000016        INTX    004176
BIT07 = 000200 G      C$ERRO= 000060      DSBINT  016674        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= 000026      EF.STA= 000040 G      F$PWR = 000017        I$CLN = 000041
BIT15 = 100000 G      C$GETW= 000027      EMAXDU  017515        F$RPT = 000012        I$DU  = 000041
BIT2  = 000004 G      C$GMAN= 000043      EN    = 000000        F$SEG = 000003        I$HRD = 000041
BIT3  = 000010 G      C$GPHR= 000042      ENAINT  016642        F$SOFT= 000005        I$INIT= 000041
BIT4  = 000020 G      C$GPLO= 000030      ENVIRN  021352        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  020716 G      I$SEG = 000041
BRINIT  004355        C$OPEN= 000034      ERRK    017474        GETSEL  021000 G      I$SETU= 000041
BSEL0 = 000000        C$PNTB= 000014      ERRLO   002204 G      G$CNTO= 000200        I$SFT = 000041
BSEL1 = 000001        C$PNTF= 000017      ERRNO = 002120        G$DELM= 000372        I$SRV = 000041
CHKAMB  016534        C$PNTS= 000016      ERRVEC= 000004 G      G$DISP= 000003        I$SUB = 000041
CHKMAN  021222 G      C$PNTX= 000015      ERTABE  003330        G$EXCP= 000400        I$TST = 000041
CHKTSS  017054        C$QIO = 000377      ERTABL  003130        G$HILI= 000002        J$JMP = 000167
CKDROP  017772        C$RDBU= 000007      ESUM    017476        G$LOLI= 000001        KIPAR0= 172340
CKEMAX  017620        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$REVI= 000003      EXPBRE  016172 G      G$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  020376        C$SEFG= 000046      EXPGT2  004465        G$PRML= 000000        KIPAR6= 172354
CONFIG  020040        C$SPRI= 000041      EXPMSG  002266 G      G$RADA= 000140        KIPAR7= 172356
COUNT   002254 G      C$SVEC= 000037      EXPREC  016164 G      G$RADB= 000000        KIPDR0= 172300
CSR   = 174400        C$TPRI= 000013      EXTA    005232        G$RADD= 000040        KIPDR1= 172302
CSRADD  002154 G      DAR   = 174404      EXTEND  005230        G$RADL= 000120        KIPDR2= 172304
CTAB    003116 G      DATA    002256 G    ESEND = 002100        G$RADO= 000020        KIPDR3= 172306
CTABE   003130 G      DATAFL  014703      ESLOAD= 000035        G$XFER= 000002        KIPDR4= 172310
CTABM   003116 G      DATASC  020756      FATCHK  017720        G$YES = 000010        KIPDR5= 172312
```

SYMBOL TABLE

```
KIPDR6= 172314          LSREV   002010 G        L10057  032114          NULCR   004426          O.GET   051630
KIPDR7= 172316          LSRPT   022702 G        L10060  032302          NXM   = 004000          O.GO    047660
KTENAB  003102 G        LSSOFT  053000 G        L10061  036702          NXR     003636          O.GO1   047736
KTFLG   003100 G        LSSPC   002056 G        L10062  034110          NXRERR  005176 G        O.GO2   047742
KTINIT  021440          LSSPCP  002020 G        L10063  034302          NXRX    003675          O.HIGH  052540
KTOFF   020064          LSSPTP  002024 G        L10064  035372          NXTU    022044          O.LG  = 000010
KTON    020046          LSSTA   002030 G        L10065  035620          OFL   = 000100          O.LGCH  052133
LERRMA  002140 G        LSSW    002134 G        L10066  040420          ONEFIL= 000000          O.LGDR  046530
LERRNO= 000000          LSTEST  002114 G        L10067  037166          OSAPTS= 000000          O.LOW   052536
LISTAL= 000001          LSTIML  002014 G        L10070  037424          OSAU  = 000001          O.MOVE  050242
LOE   = 040000 G        LSUNIT  002012 G        L10071  043600          OSBGNR= 000001          O.MSK   052534
LOOPCN  002164 G        L10000  002132          L10072  041012          OSBGNS= 000001          O.ODT   046040 G
LOOPCO  012600          L10001  002144          L10073  041312          OSDU  = 000001          O.OFST  047360
LOOPFL  003114 G        L10002  005226          L10074  041620          OSERRT= 000000          O.OLD   046756
LOT   = 000010 G        L10003  011516          L10075  044574          OSGNSW= 000001          O.OP1   046762
LSACP   002110 G        L10004  011546          L10076  046036          OSPOIN= 000001          O.OP2   047026
LSAPT   002036 G        L10005  011564          L10077  045022          OSSETU= 000001          O.OP2A  047034
LSAU    022400 G        L10006  011572          L10100  052672          O.ADR1  052550          O.ORAB  046266
LSAUT   002070 G        L10007  011610          L10101  053014          O.ALL   051134          O.ORPC  046244
LSAUTO  022604 G        L10010  011626          L10102  053224          O.AS    046630          O.ORRB  046276
LSCCP   002106 G        L10011  011640          L10104  053232          O.ASC   052117          O.P     052113
LSCLEA  022660 G        L10012  011712          MEMADD  013426 G        O.ASCI  050144          O.PCS   046256
LSCO    002032 G        L10013  012062          MENASC  021171          O.BACK  047114          O.PRNT  050402
LSDEPO  002011 G        L10014  012576          MENERR  021116          O.BALL  051020          O.PROC  047760
LSDESC  003342 G        L10015  013424          MENRES  021220          O.BD    052120          O.PROM  052126
LSDESP  002076 G        L10016  013446          MESBFA  002716 G        O.BKP = 000016          O.RALL  047304
LSDEVP  002060 G        L10017  016170          MESBFN  014453          O.BKPT  047142          O.RCSR= 177560
LSDISP  053166 G        L10020  016176          MESHEA  014636          O.BRK   050450          O.RDB = 177562
LSDLY   002116 G        L10021  016204          MMVEC = 000250          O.BW    052100          O.REG   052032
LSDTP   002040 G        L10022  016216          MPR   = 174406          O.BYT   046666          O.REGT  046156
LSDTYP  002034 G        L10023  016240          MSA.FR= 000006          O.BYT1  046660          O.REM   051304
LSDU    022476 G        L10024  016266          MSA.NO= 000000          O.CAD   052102          O.RSB   051240
LSDUT   002072 G        L10025  016426          MSA.NR= 000004          O.CADV  051446          O.RSR   051210
LSDVTY  003334 G        L10026  016736          MSA.VO= 000002          O.CLGT= 000035          O.RSTT  051400
LSEF    002052 G        L10030  022330          MSGEXP  011630 G        O.CLSE  051744          O.S     052111
LSENVI  002044 G        L10031  022474          MSGLOO  012536 G        O.COMP  050304          O.SCAN  046422
LSETP   002102 G        L10032  022602          MSGSTA  012022 G        O.CR    052123          O.SEMI  046622
LSEXP1  002046 G        L10033  022656          MSGSUB  013414 G        O.CRET  046750          O.SEQ   052116
LSEXP4  002064 G        L10034  022700          MS.ATT= 000006          O.CRLF  051776          O.SNGL  046346
LSEXP5  002066 G        L10035  023142          MS.EXT= 000200          O.CRLS  052012          O.SPAC  051732
LSHARD  052640 G        L10036  023720          MS.RSD= 000001          O.CSR1  052114          O.STM = 000340
LSHIME  002120 G        L10037  023566          MS.RSF= 000020          O.CSR2  052115          O.SVR   051150
LSHPCP  002016 G        L10040  023650          MS.RST= 000010          O.CT    052572          O.SVTT  051352
LSHPTP  002022 G        L10041  024672          NBA   = 002000          O.C1    050032          O.SWCH  052542
LSHW    002124 G        L10042  024112          NEWPAS  022032          O.DCD   046376          O.T     052112
LSICP   002104 G        L10043  024304          NODEV   003062 G        O.DCDA  046754          O.TBIT  047710
LSINIT  021602 G        L10044  024504          NOINIT  004233          O.DCDB  047302          O.TBT = 000020
LSLADP  002026 G        L10045  026152          NOINTR  004117          O.DCD1  046416          O.TCLS  046320
LSLAST  053220 G        L10046  025234          NOITS   002136 G        O.DCD2  046412          O.TCSR= 177564
LSLOAD  002100 G        L10047  025554          NOMAN   021256          O.DOT   052104          O.TDB = 177566
LSLUN   002074 G        L10050  030204          NP.IR = 000200          O.DUMP  050064          O.TL    052170
LSMREV  002050 G        L10051  026476          NP.LOO= 000040          O.EFF   047470          O.TRTC  052200
LSNAME  002000 G        L10052  026762          NP.OUT= 000100          O.ERR   046366          O.TVEC= 000014
LSPRIO  002042 G        L10053  027210          NP.WRP= 000020          O.ERR1  047464          O.TYPE  051716
LSPROT  021572 G        L10054  031414          NSI     004050          O.FCHR  052544          O.UIN   052614
LSPRT   002112 G        L10055  033470          NSINIT  004305          O.FCNT  052546          O.UPC   052530
```

CZTUWAO TU80 FRONT END PRT A MACRO M1200 29-MAR-83 13:24 PAGE 132-4
SYMBOL TABLE

```
O.URO   052512          PRMSG2  015525          RECMSG  002432 G        S1.IEO= 010000          TST10I  044252
O.USP   052526          PROASC  014533          RECV    002200 G        S1.IFM= 001000          TST11I  045660
O.WB1   046674          PR1ASC  014600          REGSAV  020656          S1.IHE= 000400          TST2ID  024666
O.WDFG  052110          PST32W  003104 G        REWIND  010254 G        S1.IID= 004000          TST3ID  026133
O.WRD   046644          PUNIT   022332          RMCHBE= 000167          S1.I1R= 020000          TST4ID  030107
O.WRD1  046710          PW.D11= 000021          RMCHEN= 000200          S1.I2R= 040000          TST5ID  031377
O.WSCH  047474          PW.D13= 000022          RMMSGB= 000104          S1.PAR= 100000          TST6ID  033371
O.XXX   052106          PW.D22= 000020          RMMSGE= 000117          S2.ATI= 000010          TST7ID  036527
PASRPT  022076          PW.NOP= 000000          RMPKTB= 000020          S2.BTI= 000004          TST8ID  040262
PATCH   053214 G        PW.NO1= 000023          RMPKTE= 000027          S2.DIM= 000200          TST9ID  042070
PATDAT  020752          PW.RDE= 000024          RMR   = 010000          S2.ILW= 000100          TTIBFR= 177562 G
PC.ERA= 002400          PW.RDR= 000001          RWPACK  010350          S2.INR= 000020          TTICSR= 177560 G
PC.IER= 002000          PW.RDS= 000005          SC    = 100000          S2.OUT= 000040          TTIVEC= 000060 G
PC.NOO= 001000          PW.RFI= 000003          SCE   = 020000          S2.UND= 000003          TTOBFR= 177566
PC.REL= 000000          PW.WCT= 000006          SCME    004711          TBLEND= 003030 G        TTOCSR= 177564
PC.REW= 000400          PW.WFI= 000004          SDELAY  010150          TCOASC  006012          TUV2A   002000 G
PKBCNT= 000006          PW.WFM= 000007          SEEK  = 000006          TCOCOD  006212          T$ARGC= 000003
PKHI  = 000004          PW.WMI= 000010          SELASC  021164          TEMP1   003064 G        T$CODE= 001130
PKLOW = 000002          PW.WNP= 000011          SELDAT= 000004          TEMP2   003066 G        T$ERRN= 002120
PKTADD  007110          PW.WTR= 000002          SEL2  = 000002          TERCLS= 000016          T$EXCP= 000000
PKTFRM  007052          P.ACK = 100000          SETMAP  020106          TESTNO= 000013          T$FLAG= 000040
PKTGET  011550 G        P.CMD = 000037          SETU    022130          TEXASC  005751          T$FREE= 053232
PKTMES  011574 G        P.CONT= 000012          SFFMSG  011566 G        TFCASC  006053          T$GMAN= 000000
PKTNEW  007145          P.CVC = 040000          SFHERR  003603          TIMEXP  016242 G        T$HILI= 000007
PKTRAM  004643 G        P.FMT = 000140          SFIERR  003550          TIMSGO  016270          T$LAST= 000001
PKTSSR  011520 G        P.FORM= 000011          SFIMSG  011506 G        TINERR  011473          T$LOLI= 000000
PNT   = 001000 G        P.GETS= 000017          SFPTBL  002134 G        TKB   = 177562          T$LSYM= 010000
PRAMPK  013450          P.IE  = 000200          SIFLAG  003106 G        TKS   = 177560          T$LTNO= 000013
PRBEXP  016160          P.INIT= 000013          SIMSG   011440          TMPBFR  002576 G        T$NEST= 000000
PRBMSG  016026          P.MODE= 007400          SKIPT   003332          TNAM    017422          T$NS0 = 000000
PRBREC  016162          P.OPP = 020000          SOFINI  016464 G        TPB   = 177566          T$NS1 = 000005
PRBTOT  016113          P.POSI= 000010          SPACE   007752 G        TPS   = 177564          T$NS2 = 000002
PRBYTE  015612 G        P.READ= 000001          SPM1    053014          TRANST  002134 G        T$NS3 = 000003
PRI   = 002000 G        P.SWB = 010000          SPM4    053053          TSBA  = 177776 G        T$PCNT= 000000
PRIADD  007524          P.WRIT= 000005          SPM6    053103          TSBAH = 177777 G        T$PTAB= 010103
PRIAO   007574          P.WRTC= 000004          SPM7    053133          TSBAL = 177776 G        T$PTHV= 000001
PRIBXO  007156 G        P.WRTS= 000006          SR0   = 177572          TSBAM2  024524          T$PTNU= 000001
PRIEQU  007424          QVP     002152 G        SR1   = 177574          TSBAM3  024606          T$SAVL= 177777
PRIPKT  006704 G        RAMASC  013616          SR2   = 177576          TSDB  = 177776 G        T$SEGL= 177777
PRIRAM  007432          RAMDAT  002206 G        SR3   = 172516          TSDBH = 177777 G        T$SEKO= 010000
PRITAD  007640          RAMER   010456 G        SSR   = 000200          TSDBL = 177776 G        T$SIZE= 000005
PRITSS  005264          RAMERR  016200 G        STATCO  012064          TSFCOD  006552          T$SUBN= 000001
PRITO   007710          RAMEXP  016220 G        SVCGBL= 000000          TSREJ = 000006          T$TAGL= 177777
PRIXOR  007306 G        RAMFHR  014362          SVCINS= 000000          TSSDEF  006122          T$TAGN= 010105
PRI00 = 000000 G        RAMFOR  007462          SVCSUB= 000001          TSSR  = 000000 G        T$TEMP= 000014
PRI01 = 000040 G        RAMHLD  010640          SVCTAG= 000000          TSSRBI  003400 G        T$TEST= 000013
PRI02 = 000100 G        RAMIOP  010644          SVCTST= 000001          TSSRFO  005731          T$TSTM= 177777
PRI03 = 000140 G        RAMPD   010715          S$LSYM= 010000          TSSRH = 000001 G        T$TSTS= 000001
PRI04 = 000200 G        RAMRSH  010642          S0.IDB= 000010          TSSX    003716          T$$AU = 010031
PRI05 = 000240 G        RAMSIZ  002246 G        S0.IFB= 000002          TSTBLK  002720 G        T$$AUT= 010033
PRI06 = 000300 G        RAMTAD  016206 G        S0.IFP= 000001          TSTCNT  002162 G        T$$CLE= 010034
PRI07 = 000340 G        RBPCRA  014750          S0.ILD= 000020          TSTEND  017436          T$$DAT= 010104
PRMESS  013702          RCVHIA  002250 G        S0.ION= 000040          TSTFLA  002260 G        T$$DU = 010032
PRMNO   002264 G        RCVLOA  002252 G        S0.IRD= 000100          TSTLOO  017174 G        T$$HAR= 010100
PRMSGE  015242 G        RDERR   005104          S0.IRW= 000004          TSTPTR  002262 G        T$$HW = 010000
PRMSGO  015422          READ  = 000014          S0.ISP= 000200          TSTSET  017226 G        T$$INI= 010030
                        READY = 000001          S1.ICE= 002000          TST1ID  023700           T$$MSG= 010025
```

SYMBOL TABLE

```
T$$PC = 000001          T3.2    025250          T7BUFR  035714          T9LOAD  041724          XSOCON  015015
T$$PRO= 010027          T4      026154 G        T7DATA  035640          T9LOOP  040456          XSOEOT= 000001
T$$PTA= 010103          T4BFR   027264          T7DTA   035702          T9MSGB  042261          XSOIE = 000040
T$$RPT= 010035          T4DATA  027250          T7INT   036440          T9NINT  042445          XSOILA= 000400
T$$SEG= 010000          T4INT   027735          T7LOOP  033526          T9NXM   042627          XSOILC= 001000
T$$SOF= 010101          T4LOOP  026210          T7MBF   035734          T9PACK  041660          XSOLET= 020000
T$$SRV= 010026          T4NBA   027376          T7NBA   036031          T9PAR6  041726          XSOMOT= 000200
T$$SUB= 010077          T4PACK  027240          T7NINT  036347          T9SETG  043474          XSONEF= 002000
T$$SW = 010001          T4REST  030136          T7NNBA  036113          T9SWRT  043426          XSOONL= 000100
T$$TES= 010076          T4SP    027260          T7PACK  035630          T9TBE   042066          XSOPED= 000010
T1      023464 G        T4SSR   027646          T7PKT   035672          T9WRTS  042172          XSORLL= 010000
T1LOOP  023520          T4TSBA  030024          T7RST   036556          T9.1    040456          XSORLS= 040000
T1.1    023522          T4.1    026210          T7RT2   036630          T9.2    041026          XSOTMK= 100000
T1.2    023602          T4.2    026512          T7SSR   036260          T9.3    041326          XSOVCK= 000020
T10     043602 G        T4.3    026764          T7SSRM  036170          T91LOO  040600          XSOWLE= 004000
T10BFR  044232          T42DAT  027304          T7.1    033526          T92LOO  041144          XSOWLK= 000004
T10DAT  044220          T42DON= 027320          T7.2    034112          T93LOO  041374          XS1CON  015062
T10LOO  043636          T42NBA  027320          T7.3    034504          T94TST  041732          XS2CON  015127
T10MEM  044325          T42REJ  027451          T7.4    035374          UAM   = 000200 G       XS3CON  015174
T10NBA  044364          T44REJ  027550          T8      036704 G        UNITN   002150 G       XXCOMM  003070 G
T10PAC  044210          T5      030206 G        T8BFR   037462          UNREC = 000006          X$ALWA= 000000
T10RES  044526          T5BFR   030732          T8BF2   037532          USI     004021          X$FALS= 000040
T10SSR  044437          T5DATA  030720          T8DATA  037450          WAITF   016740 G        X$OFFS= 000400
T11     044576 G        T5LOOP  030242          T8DTA   037520          WC.IFA= 000200          X$TRUE= 000020
T11BFR  045066          T5NMSG  031300          T8LOOP  036740          WC.IFE= 000002          X1.COR= 020000
T11BS0  045060          T5NVCK  031121          T8NBA   037552          WC.IGO= 000001          X1.DLT= 100000
T11BS1  045061          T5PACK  030710          T8NINT  040134          WC.IRE= 000010          X1.MBZ= 017375
T11BS2  045062          T5SSR   031211          T8PACK  037440          WC.IRW= 000004          X1.RBP= 000400
T11DAT  045060          T5VCK   030752          T8PK2   037510          WC.IOT= 000100          X1.SPA= 040000
T11DTA  045500          T5VCK2  031045          T8REST  040314          WC.I1T= 000040          X1.UNC= 000002
T11LOO  044632          T6      031416 G        T8RT2   040356          WC.I5R= 000020          X2.BUF= 000100
T11NBA  045512          T6BFR   032342          T8SR2   040060          WF.IED= 000010          X2.EXT= 000200
T11NIN  045566          T6DATA  032330          T8SSR   040004          WF.IER= 000004          X2.OPM= 100000
T11PAC  045050          T6INT   033217          T8TSBA  040212          WF.IHI= 000200          X2.RCE= 040000
T11PK2  045470          T6LOOP  031452          T8.1    036740          WF.IRE= 000040          X2.REV= 000077
T11RES  045726          T6NBA   032376          T8.2    037202          WF.IWF= 000020          X2.SPA= 035400
T11RST  045764          T6NINT  033126          T82REJ  037612          WF.IWR= 000100          X2.UNI= 000007
T11.1   044632          T6PACK  032320          T83REJ  037673          WF.I3R= 000002          X2.WCF= 002000
T2      023722 G        T6REST  033416          T84REJ  037723          WF.I4R= 000001          X3.DCK= 000010
T2LOOP  023760          T6SSR   033037          T9      040422 G        WRTCHR  010152 G        X3.MBZ= 000006
T2.1    023740          T6TSBA  033306          T9BFR   041702          WRTERR  005011          X3.MDE= 177400
T2.2    024114          T6.1    031452          T9BKGN  042357          WRTMSG  004754          X3.OPI= 000100
T2.3    024306          T6.2    031704          T9BLK   041734          XFERAS  016430          X3.REV= 000040
T3      024674 G        T6.3    032116          T9CHAR  043536          XNXM    017114          X3.RIB= 000001
T3INT   025723          T62DAT  032362          T9CKRA  043316 G        XORBFO  007240          X3.SPA= 000200
T3LOOP  024730          T62DON= 032376          T9CONV  042736          XORFOR  007356          X3.TRF= 000020
T3NBA   025620          T62REJ  032451          T9CT2   043130          XST0  = 000006 G        X4.HSP= 100000
T3NINT  026001          T63REJ  032550          T9DATA  041670          XST1  = 000010 G        X4.MBZ= 017400
T3PACK  025610          T64REJ  032643          T9DPR   042536          XST2  = 000012 G        X4.RCE= 040000
T3SSR   025645          T65REJ  032741          T9GETS  042116          XST3  = 000014 G        X4.TSM= 020000
T3TSBA  026061          T7      033472 G        T9HIAD  041722          XST4  = 000016 G        X4.WRC= 000377
T3.1    024730          T7BFR   035652          T9KT    041730          XSOBOT= 000002

. ABS.  053232    000
        000000    001
ERRORS DETECTED:  0
```

CZTUWA0 TU80 FRONT END PRT A MACRO M1200 29-MAR-83 13:24 PAGE 132-6
SYMBOL TABLE

VIRTUAL MEMORY USED: 36168 WORDS (142 PAGES)

DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:08:01
CZTUWA.BIC,CZTUWA/-SP=SVC.MLB/ML,CZTUWA