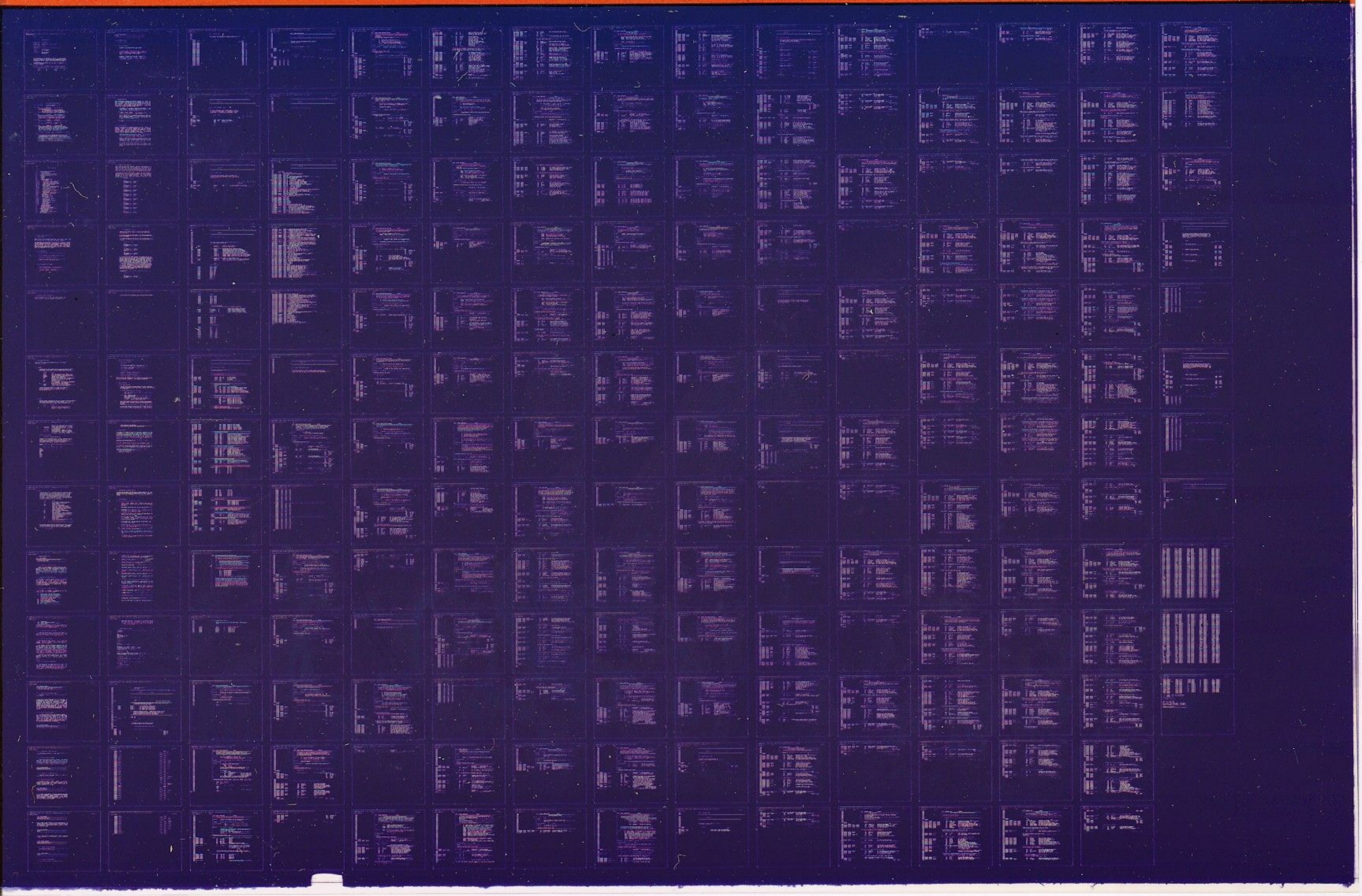
DHV11-M

DHV11-M FUNC TST PART 1
CVDHAD0

AH-T653D-MC
1 OF 1 OCT 1985

digital
MADE IN USA



```
.REM &
```

IDENTIFICATION
------ -------

PRODUCT CODE: AC-T652D-MC

PRODUCT NAME: CVDHADO DHV11-M FUNC TST PART 1

PRODUCT DATE: MAY 1985

MAINTAINER: Bruce Ribolini - MK Diagnostics Group

AUTHOR: Bert Kleinschmidt
Tony Grimshaw

MODIFIED BY: Bert Kleinschmidt
Anthony Hart
Peter ONeil

```
                        ***** MODIFICATION HISTORY *****

Original release:       31 OCT-83 (EDITED 11-JUL-83)
                        Bert Kleinschmidt

Version B0              09-OCT-83       Bert Kleinschmidt
        Fixed typographical errors.
        Moved test from CVDHB (Part 2) into this program:
                Old CVDHB (version A) tests 2 through 8 are
                now New CVDHA (version B) tests 20 through 26.
        Modified test 13 (Selftest Fail Test) to allow use of CVDHA
                in manufacturing with version 0 DHV ROM code.

Version C0              17-JUL-84       Bert Kleinschmidt
        Modified control of processor priority throughout the
        program to maintain priorities of 5 or lower to allow
        LTC interrupts.  This guarantees a less than 2 second
        responce to a Break request while running under APT.

 Version C0              9-SEP-84        PETER ONEIL
        Modified cleanup code to turn off clock if it was turned on.

                        29-MAR-85       Howard Marshall
        Did formal release of version CVDHAC0.

Version D0              8-May-85        Howard Marshall
        Found timing problems when running under the extended
        DRS monitor on a PDP 11/23+ in tests: #10, Skip selftest
        test and #13, Selftest fail test because extra under slower
        CPUs, DRS calls consume too much time and interfere with
        the critical timing loops in these tests. Substituted
        SETPRI + GETPRI with MTPS + MFPS respectively.

        ATTENTION: /NO RESTORE (of DRS) IS ASSEMBLED INTO THIS DIAG. !!!
                                )
                                V
      ************************************************************************
      *  Found additional timing problems when runnirg under the            *
      * extended DRS monitor on a PDP 11/23+ in test #3, Master reset        *
      * skip selftest test. The DRS calls are consuming too much time        *
      * and adversly affecting the timing of the critical loops in test      *
      * #3. Assembled in "/NORESTORE" into this source code by adding        *
      * an argument (a 1) to the end of the HEADER macro.                    *
      ************************************************************************
```

TABLE OF CONTENTS

1.0 GENERAL PROGRAM CONSIDERATIONS

1.1 PROGRAM ABSTRACT

CVDHA is part one of the DHV11-M functional verification test. This part of the test verifies that the reset, register access, and interrupt functions of the board are functioning correctly.

THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP+, ACT, APT, SLIDE AND PAPER TAPE. FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP+ USER'S MANUAL. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN THE OPERATING INSTRUCTIONS-COMMANDS OF THIS DOCUMENT.

1.2 SYSTEM REQUIREMENTS

The following hardware is required to run the DHV FVT:

- o LSI-11 processor with at least 32 Kbytes of RAM.
- o DHV11 boards installed on the Q-bus.
- o Appropriate program load device supporting XXDP+ media or a down-line loading system.

1.3 RELATED DOCUMENTS AND STANDARDS

- o DHV11-M Hardware Manual - This manual describes the functions and uses of the DHV11-M device.
- o XXDP+ User's Manual - Describes the running of diagnostics under the XXDP+ monitor.

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

The LSI-11 processor, the Q-BUS, the system memory, the console terminal, and the load media are assumed to have been tested and found working before this program 's run.

PROGRAM DOCUMENT

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES. FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL CHQUS).

2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES (SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER ↑C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SEE PERFOMANCE AND PROGRESS REPORTS SECTION OF THIS DOCUMENT)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE FLAGS SECTION)
ZFLAGS	CLEAR ALL FLAGS (SEE FLAGS SECTION)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".
MORE INFORMATION CAN BE FOUND WITHIN THE SECTION LABELLED EXTENDED COMMAND SYNTAX

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH.
IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10.

```
                                THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO
                                BE RUN.  ALL OTHER TESTS WILL NOT BE RUN.
                /PASS:DDDDD     EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
                /FLAGS:FLGS     SET SPECIFIED FLAGS.SEE THE FLAGS SECTION
                                OF THIS DOCUMENT.
                /EOP:DDDDD      REPORT END OF PASS MESSAGE AFTER EVERY
                                DDDDD PASSES ONLY.  (DDDDD = 1 TO 64000)
                /UNITS:LIST     TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED
                                IN THE LIST.  LIST EXAMPLE - /UNITS:0:5:10-12
                                USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

        START/TESTS:1-5/PASS:1000/EOP:100

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE
EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF
PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY.  A
SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS.  YOU MAY,
FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH
COMMAND.

          TESTS   PASS    FLAGS   EOP     UNITS
          ------- ------------------------
START       X       X       X       X       X
RESTART     X       X       X       X       X
CONTINUE            X       X       X
PROCEED                     X
DROP                                        X
ADD                                         X
PRINT
DISPLAY                                     X
FLAGS
ZFLAGS
EXIT
```

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXR*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST
EVL	EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)

*SEE THE ERROR INFORMATION SECTION OF THIS DOCUMENT.

SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A "BELL" ON ERROR, YOU MAY USE THE FOLLOWING STRING:

```
/FLAGS:LOE:IER:BOE
```

2 4 EXTENDED COMMAND SYNTAX

2.4.1 START COMMAND -

```
*****************************************************************
STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
   <FLAG-LIST>/EOP:<INCR>
*****************************************************************
```

2.4.1.1 Tests Switch (/TESTS:<TEST-LIST>) -

<TEST-LIST> Is a sequence of decimal numbers (1:2 etc.) or ranges of decimal numbers (1-5:8-10 etc.), seperated by colons, that specify the tests to be executed.Tests will be executed in numerical order regardless of the order of specification. The default is to execute all tests. On this and all switches, the angle brackets <> are punctuation used in the definition only, and are not to be typed by the operator. See example at end of "Effect of Start Command" section.

2.4.1.2 Pass Switch (/PASS:<PASS-CNT>) -

<PASS-CNT> Is a decimal number indicating the desired number of passes. A pass is defined as the execution of the full diagnostic (all selected tests).The default is non-ending execution. In this case, exit from the program is accomplished either by typing a control/C or by occurance of an error with the halt on error flag being set. The exit is a return to command mode. See example at end of "Effect of Start Command" section.

2.4.1.3 Flags Switch (/FLAGS:<FLAG-LIST>) -

<FLAG-LIST> is a sequence of elements of the form <FLAG>, <FLAG=1>, or <FLAG=0>, separated by colons, where <FLAG> has one of the following values:

HOE	Halt on error, causing command mode to be entered when an error is encountered.
LOE	Loop on error, causing the diagnostic to loop continuously within the smallest defined block of coding (segment, subtest, or test) containing the error.
IER	Inhibit error reporting.
IBE	Inhibit basic error reports.
IXE	Inhibit extended error reports.
PRI	Direct all messages to a line printer.
PNT	Print number of test being executed.
BOE	Bell on error.
UAM	Run in unattended mode, bypassing manual

intervention.
ISR Inhibit statistical reports.
IDU Inhibit dropping of units by diagnostic.
LOT Loop on test.
The flags named or equated to 1 are set, those equated to 0 are cleared. A flag not specified is cleared. If the flags switch is not given all flags are cleared. See example at end of "Effect of Start Command" section.

2.4.1.4 End Of Pass Switch (/EOP:<INCR>) -

<INCR> Is a decimal number indicating how often (in terms of passes) it is desired that the end of pass message be printed. The default is at the end of every pass. See example at end of "Effect of Start Command" section.

2.4.1.5 Effect Of Start Command -

The effect of the start command is to initiate the hardware parameter dialogue, the software parameter dialogue, the initialization questions, and then the diagnostic commences testing.

The hardware parameter dialogue commences with the question "# UNITS (D) ?" to which the operator should reply with the number of units to be tested. Following this are the questions whereby the P-Tables themselves are built. Each P-Table is a core-resident table containing all the hardware information for one complete unit.Each question is followed by the response radix (D for decimal, B for binary, O for octal, L for Yes/No) in parentheses and the default value after the parentheses. For the actual Hardware P Table questions see the "Hardware Parameters" section.

Following the hardware questions are the software questions to build the software tables, which define operating parameters of the diagnostic program. These Questions are described in the "Software Parameters" section.

EXAMPLE:

```
STA/TESTS:1:3-4:/PASS:3/FLAGS:IER:HOE=1
```

This command will cause three passes to be made, with each pass consisting of tests 1,3, and 4. There is no difference between saying <FLAG> and saying <FLAG=1>. The notation <FLAG=0> is meaningful only on a command other than start to clear a flag that was previously set. Note that on all commands only the first three letters are scanned.

2.4.2 Restart Command -

```
**************************************************************
RES(TART)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
   <FLAG-LIST>/UNITS:<UNIT-LIST>
**************************************************************
```

2.4.2.1 Tests, Pass, And Flags Switches -

<TEST-LIST>, <PASS-CNT>, and <FLAG-LIST> are as in the start command.

2.4.2.2 Units Switch (/UNITS:<UNIT-LIST>) - <UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIAGLOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

2.4.2.3 Effect Of Restart Command -

The restart command differs from the start command in that the P-Tables from the previous start command (there must have been one) are used, instead of new ones being built. The software dialogue may optionally be reexecuted (operator will be asked). The command can be used after command mode has been reentered in any of the three normal ways: a) the requested number of passes have been made, b) an error was encountered with the halt on error flag set, or c) a control/C was entered by the operator.

2.4.3 Continue Command -

```
**************************************************************
CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>
**************************************************************
```

2.4.3.1 Flag Switch (/FLAGS:<FLAG-LIST>) -

<FLAG-LIST> Is same as in the start command, but unspecified flags retain their current value.

2.4.3.2 Effect Of Continue Command -

Continue must follow a start or restart, and command mode must have been entered due to a halt on error or a control/C. The effect of the command is to go to the beginning of the test that was being executed when the halt or control/C took place. Software dialogue may optionally be reexecuted. Hardware parameters may not be changed.

2.4.4 Proceed Command -

****
PRO(CEED)/FLAGS:<FLAG-LIST>
****

2.4.4.1 Flags Switch (/FLAGS:<FLAG-LIST>) -

<FLAG-LIST> Is as in the start command, but unspecified flags retain their current value.

2.4.4.2 Effect Of Proceed Command -

Proceed must follow a start, restart, or continue. Command mode must have been entered via a halt on error. The effect of the command is to begin execution at the location following the error call. Neither hardware nor software parameters may be altered.

2.4.5 Add Command -

****
ADD/UNITS:<UNIT-LIST>
****

2.4.6 EFFECT OF ADD COMMAND - THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

PROGRAM DOCUMENT

2.4.7 Drop Command -
****
DRO(P)/UNITS:<UNIT-LIST>
****

2.4.8 EFFECT OF DROP COMMAND - THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

2.4.9 Print Command -
****
PRI(NT)
****

2.4.9.1 Effect Of Print Command - Error summary reporting is not implemented in this diagnostic, so this command has no effect.

2.4.10 Display Command -
****
DIS(PLAY)/UNITS:<UNIT-LIST>
****

2.4.10.1 Effect Of Display Command -

The hardware P-Tables for all units are printed in the format in which they were entered.

2.4.11 Flags Command -
****
FLA(GS)
****

2.4.11.1 Effect Of Flags Command -

The current settings of all flags are printed.

PROGRAM DOCUMENT

2.4.12 Zflags Command -

****
ZFL(AGS)
****

2.4.13 Zflags Command

All flags are cleared.

2.4.14 Control Characters -

C A control/C (C) entered during the execution of a diagnostic causes a return to command mode.

Z A control/Z (Z) entered during one of the two operator dialogues hardware P-Table dialogue or software P-Table dialogue causes the defaults to be taken for the remainder of that dialogue.

O A control/O (O) entered during the execution of a diagnostic causes all teletype output to be surpressed for the remainder of the diagnostic or until another control/O is typed, which restores normal teletype output.

2.5 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?" YOU MUST ANSWER "Y" AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 6 OF THE XXDP+ USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A "Y", THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL). YOU WILL THEN BE ASKED THE FOLLOWING QUESTIONS FOR EACH UNIT:

1. CSR ADDRESS - This question requests the CSR address of the specified DHV11. The default answer for this question is the lowest address in the PDP-11 floating address space in which a DHV11-M can be placed (160460 Octal).

2. INTERRUPT VECTOR ADDRESS - This question requests the interrupt vector address of the specified DHV11.

3. ACTIVE LINES BIT MAP - This question requests an octal bit map of the serial communication lines on the DHV11 which are being selected for testing. If the bit in the bit map is set which corresponds to a particular line (i.e. bit 3 for line 3) that line will be tested by the FVT.

4. BR Level This questions requests the interrupt BR level of the DHV11.

2.6 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". The following Software P-Table questions are asked by the program if the operator indicates that the Software Parameters are to be changed:

1. REPORT UNIT NUMBER AS EACH UNIT IS TESTED - This question asks whethe the program should report the number of the unit which it is testing as it begins to test each unit.

2. NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE - This question asks for the number of data errors which should be reported individually by this program for each line for each transmission test. Errors which are not reported individually are reported in summary error reports.

3. ROM VERSION PRINTOUT ON THE FIRST PASS - This question asks whether the program should print out the versions of the on board 8051 processor ROMs dur'ng the first pass of the program.

2.7 EXTENDED P-TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES IN A TABLE THAT DESCRIBES THE DEVICES UNDER TEST. THE SIMPLEST WAY TO BUILD THIS TABLE IS TO ANSWER ALL QUESTIONS FOR EACH UNIT TO BE TESTED. IF YOU HAVE A MULTIPLEXED DEVICE SUCH AS A MASS STORAGE CONTROLLER WITH SEVERAL DRIVES OR A COMMUNICATION DEVICE WITH SEVERAL LINES, THIS BECOMES TEDIOUS SINCE MOST OF THE ANSWERS ARE REPETITIOUS.

TO ILLUSTRATE A MORE EFFICIENT METHOD, SUPPOSE YOU ARE TESTING A FICTIONAL DEVICE, THE XY11. SUPPOSE THIS DEVICE CONSISTS OF A CONTROL MODULE WITH EIGHT UNITS (SUB-DEVICES) ATTACHED TO IT. THESE UNITS ARE DESCRIBED BY THE OCTAL NUMBERS 0 THROUGH 7. THERE IS ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS CALLED THE Q-FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

```
# UNITS (0) ?  8<CR>

UNIT 1
CSR ADDRESS (0) ?  160000<CR>
SUB-DEVICE # (0) ?  0<CR>
Q-FACTOR (0) 0 ?  1<CR>

UNIT 2
CSR ADDRESS (0) ?  160000<CR>
SUB-DEVICE # (0) ?  1<CR>
Q-FACTOR (0) 1 ?  0<CR>

UNIT 3
CSR ADDRESS (0) ?  160000<CR>
SUB-DEVICE # (0) ?  2<CR>
Q-FACTOR (0) 0 ?  <CR>

UNIT 4
CSR ADDRESS (0) ?  160000<CR>
SUB-DEVICE # (0) ?  3<CR>
Q-FACTOR (0) 0 ?  <CR>

UNIT 5
CSR ADDRESS (0) ?  160000<CR>
SUB-DEVICE # (0) ?  4<CR>
Q-FACTOR (0) 0 ?  <CR>

UNIT 6
CSR ADDRESS (0) ?  160000<CR>
SUB-DEVICE # (0) ?  5<CR>
Q-FACTOR (0) 0 ?  <CR>

UNIT 7
CSR ADDRESS (0) ?  160000<CR>
SUB-DEVICE # (0) ?  6<CR>
Q-FACTOR (0) 0 ?  1<CR>

UNIT 8
CSR ADDRESS (0)  160000<CR>
SUB-DEVICE # (0) ?  7<CR>
```

```
        Q-FACTOR (0) 1 ?  <CR>
```

NOTICE THAT THE DEFAULT VALUE FOR THE Q-FACTOR CHANGES WHEN A NON-DEFAULT RESPONSE IS GIVEN. BE CAREFUL WHEN SPECIFYING MULTIPLE UNITS!

AS YOU CAN SEE FROM THE ABOVE EXAMPLE, THE HARDWARE PARAMETERS DO NCT VARY SIGNIFICANTLY FROM UNIT TO UNIT. THE PROCEDURE SHOWN IS NOT VERY EFFICIENT.

THE RUNTIME SERVICES CAN TAKE MULTIPLE UNIT SPECIFICATIONS HOWEVER. LET'S BUILD THE SAME TABLE USING THE MULTIPLE SPECIFICATION FEATURE.

```
        # UNITS (D) ?  8<CR>

        UNIT 1
        CSR ADDRESS (O) ?  160000<CR>
        SUB-DEVICE # (O) ?  0,1<CR>
        Q-FACTOR (O) 0 ?  1,0<CR>

        UNIT 3
        CSR ADDRESS (O) ?  160000<CR>
        SUB DEVICE # (O) ?  2-5<CR>
        Q-FACTOR (O) 0 ?  0<CR>

        UNIT 7
        CSR ADDRESS (O) ?  160000<CR>
        SUB-DEVICE # (O) ?  6,7<CR>
        Q-FACTOR (O) 0 ?  1<CR>
```

AS YOU CAN SEE IN THE ABOVE DIALOGUE, THE RUNTIME SERVICES WILL BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. IN THE FIRST PASS, TWO ENTRIES ARE BUILT SINCE TWO SUB-DEVICES AND Q-FACTORS WERE SPECIFIED. THE SERVICES ASSUME THAT THE CSR ADDRESS IS 160000 FOR BOTH SINCE IT WAS SPECIFIED ONLY ONCE. IN THE SECOND PASS, FOUR ENTRIES WERE BUILT. THIS IS BECAUSE FOUR SUB-DEVICES WERE SPECIFIED. THE "-" CONSTRUCT TELLS THE RUNTIME SERVICES TO INCREMENT THE DATA FROM THE FIRST NUMBER TO THE SECOND. IN THIS CASE, SUB-DEVICES 2, 3, 4 AND 5 WERE SPECIFIED. (IF THE SUB-DEVICE WERE SPECIFIED BY ADDRESSES, THE INCREMENT WOULD BE BY 2 SINCE ADDRESSES MUST BE ON AN EVEN BOUNDARY.) THE CSR ADDRESSES AND Q-FACTORS FOR THE FOUR ENTRIES ARE ASSUMED TO BE 160000 AND 0 RESPECTIVELY SINCE THEY WERE ONLY SPECIFIED ONCE. THE LAST TWO UNITS ARE SPECIFIED IN THE THIRD PASS.

THE WHOLE PROCESS COULD HAVE BEEN ACCOMPLISHED IN ONE PASS AS SHOWN BELOW.

```
        # UNITS (D) ?  8<CR>

        UNIT 1
        CSR ADDRESS (O) ?  160000<CR>
        SUB DEVICE # (O) ?  0-7<CR>
        Q-FACTOR (O) 0 ?  0,1,0,...,1,1<CR>
```

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING
A NULL FIELD) TELL THE RUNTIME SERVICES TO REPEAT THE LAST REPLY.

2.8 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK AND THE QUESTION IS ASKED) QUESTIONS
3. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM
4. TYPE "START"
5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. FOR DEFAULT INFORMATION SEE THE SECTIONS WITHIN THIS DOCUMENT ON FLAGS,AND HARDWARE QUESTIONS.

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY A DIAGNOSTIC: GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES ARE ALWAYS PRINTED UNLESS THE "IER" FLAG IS SET (SEE THE FLAGS SECTION OF THIS DOCUMENT).

THE GENERAL ERROR MESSAGE IS OF THE FORM:

```
NAME  TYPE- NUMBER  ON UNIT NUMBER  TST NUMBER PC:XXXXXX
ERROR MESSAGE
```

```
.WHERE; NAME = DIAGNOSTIC NAME
        TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
        NUMBER = ERROR NUMBER
        UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)
        TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
        PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL
```

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS THE "IER" OR "IBR" FLAGS ARE SET (SEE THE FLAGS SECTION OF THIS DOCUMENT).
THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBR" OR "IXR" FLAGS ARE SET (SEE THE

FLAGS SECTION OF THIS DOCUMENT).
THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR
MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 ERROR MESSAGES

This program is intended to provide a go/no-go indication of the functionality of DHV11-M boards. To execute the program in this mode the operator can run with the inhibit basic error reporting switch. In this mode the program prints error messages which contain the error message header described above, plus the name of the failing test. For a list of the test names in this program see the test summaries section of this document. An example of such an error message is the following:

```
CVDHA DVC FTL ERR 01603 ON UNIT 02 TST 015 SUB 000 PC: 015244
DEVICE REGISTER WORD READ/WRITE TEST
```

This error indicates that a fatal error was encountered within the test which tests the read/write capability of the DHV11-M registers.

If the operator requires more extensive error reporting he can run with all error reporting enabled by not using the inhibit reporting switches. The above error message would then become the following:

```
CVDHA DVC FTL ERR 01603 ON UNIT 02 TST 015 SUB 000 PC: 015244
DEVICE REGISTER WORD READ/WRITE TEST
  BAD BIT(S) IN DEVICE TBUFFAD1 REGISTER FOR LINE  7 (D).
    EXPECTED DATA: 000000 (0).
    ACTUAL DATA:   000023 (0).
```

4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH THE TOTAL NUMBER OF ERRORS REPORTED SINCE THE DIAGNOSTIC WAS STARTED. THE "EOP" SWITCH CAN BE USED TO CONTROL HOW OFTEN THE END OF PASS MESSAGE IS PRINTED. FOR FUTHER INFORMATION SEE THE SWITCHES SECTION OF THIS DOCUMENT.

5.0 TEST SUMMARIES

The following tests are included within CVDHA:

1. Device register address test - Verifies that the UUT registers will respond witht the proper Q-BUS handshaking when accessed. Verifies that the UUT is at the proper address.

2. MASTER.RESET (Selftest) test - Verifies that the MASTER.RESET bit clears within a specified time of it being set.

3. MASTER.RESET (Skip Selftest) test - Verifies that the MASTER.RESET bit clears within a short time after it is set if the skip selftest sequence is used.

4. RX.CHARACTER field test - Verifies that the data bits of the codes in the FIFO after a reset and skip selftest are consistant with the skip selftest codes.

5. Reception flag field test - Verifies that the 3 data status bits (OVERRRUN, FRAMING, and PARITY error bits) are all set on all of the skip selftest codes in the FIFO after a reset and skip selftest sequence.

6. RX.DATA.AVAIL test - Verifies that the RX.DATA.AVAIL bit is set when the skip selftest codes are in the FIFO and that it clears after they are read.

7. RX.DATA.VALID test - Verifies that the RX.DATA.VALID bit is set for each valid skip selftest code in the FIFO and clear after all valid codes are read.

8. RX.LINE field test - Verifies that the RX.LINE fields are correct for the skip selftest codes.

9. BMP run test - This test runs the BMP and verifies that it does not fail within a specified period. This test should signal problems that the BMP codes could cause with later tests.

10. Skip selftest test - This test verifies that if the selftest is skipped the proper codes are placed in the FIFO and that no errors are encountered.

11. DIAGNOSTIC.FAIL (Skip selftest) test - This test verifies, by using the skip selftest sequence, that the DIAGNOSTIC.FAIL bit will go to both the active and inactive states.

12. Selftest run test - Verifies that no errors are found by the execution of the selftest.

13. Selftest fail test - Verifies that the selftest will report errors correctly when it is forced to fail.

14. ROM version printout test - If requested, reports the version numbers of the 8051 ROMs.

15. Word access read/write test - Verifies that the registers respond correctly to word read and write accesses.

16. Word access read/modify/write test - Verifies that the registers respond correctly to read/modify/write word accesses.

17. Byte access read/write test - Verifies that the registers respond correctly to byte read and write accesses.

18. Byte access read/modify/write test - Verifies that the registers respond correctly to read/modify/write byte accesses.

19. ID.BIT test - Verifies that the ID.BIT reads as a zero.

20. No TX.DATA.VALID/No TX.ACTION test - Verifies that if a data word is written without the TX.DATA.VALID bit set, no TX.ACTION is generated. This test does not require that characters are TXed.

21. TX.DATA.VALID / TX.ACTION test - Verifies that if a data word is written with the TX.DATA.VALID bit set, it generates a corresponding TX.ACTION. This test does not require that characters are TXed.

22. TX.ENABLE inactive test - Verifies that if the TX.ENABLE bit is clear no transmission occurs.

23. TX.ENABLE active test - Verifies that TX occurs if the TX.ENABLE is set.

24. Interrupts test - Verifies that the TX and RX interrupts are functioning correctly.

25. BR level test - Verifies that the UUT generates TX and RX interrupts at the correct BR level.

26. DIAG field (BMP) test - Verifies that a request for BMP code reporting is answered by the UUT within the specified time.

27. Report BMP codes test - This pseudo test reports the first 32 BMP codes which were discovered in the FIFO during the execution of the other tests. This avoids the interruption of other tests by these codes, if they are not critical to the tests being performed.

6.0 EXAMPLE ERROR FREE PASS

The following is an example of an error free pass dialogue:

```
.R CVDHADO
CVDHADO.BIC

DRS
CVDHA-D-0
DHV11-M FUNC TST PART1
UNIT IS DHV11-M
RESTART ADDR: 147670
DR>STA

CHANGE HW (L)  ? Y

# UNITS (D)  ? 2

UNIT 0
CSR ADDRESS:  (O)  160460 ? ↑Z

UNIT 1
CSR ADDRESS:  (O)  160460 ? 160040
INTERRUPT VECTOR ADDRESS:  (O) 300 ? 320
ACTIVE LINE BIT MAP:  (O)  377 ? <CR>
INTERRUPT BR LEVEL:  (O)  4? <CR>

CHANGE SW (L)  ? Y

REPORT UNIT NUMBER AS EACH UNIT IS TESTED:  (L) Y ? <CR>
NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE:  (D)  0 ? 4
ROM VERSION PRINTOUT ON THE FIRST PASS:  (L) Y ?

 TESTING UNIT :  0(D)

 ROM VERSION NUMBERS:  PROC_1 =  2(D)   PROC_2 =  2(D)

 TESTING UNIT :  1(D)

 ROM VERSION NUMBERS:  PROC_1 =  2(D)   PROC_2 =  2(D)

CVDHA EOP    1
    0 CUMULATIVE ERRORS

 TESTING UNIT :  0(D)
↑C
DR> EXIT
```

```
                                &
                                        .LIST SEQ,LOC,BIN,MEB
                                        .NLIST CND

                                ;*****************************************************************************************************
                                ;
                                ;               VDHA.PHD
                                ;
                                ;*****************************************************************************************************



                                .SBTTL  Program Header


                                        .MCALL  SVC
000000                                  SVC                                     ; INITIALIZE SUPERVISOR MACROS

                                ;*****************************************************************************************************
                                ;       IF STRUCTURED MACROS ARE TO BE USED, ADD ".MCALL STRUCT" AND "STRUCT"
                                ;       TO INITIALIZE THE STRUCTURED MACROS.

000001                          SVCINS= 1       ; LIST INSTRUCTIONS, SHIFTED RIGHT
000001                          SVCTST= 1       ; LIST TEST TAGS, SHIFTED RIGHT
000001                          SVCSUB= 1       ; LIST SUBTEST TAGS, SHIFTED RIGHT
000001                          SVCGBL= 1       ; LIST GLOBAL TAGS, SHIFTED RIGHT
000001                          SVCTAG= 1       ; LIST OTHER TAGS, SHIFTED RIGHT

                                ;       CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
                                ;       TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
                                ;       SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
                                ;       CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
                                ;*****************************************************************************************************

000000                          .ENABL  ABS
                                ;.ENABL AMA
002000                          .       =       2000

002000                                  BGNMOD

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

002000                                  POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL


                                        HEADER  CVDHA,D,0,16,0,PRI07,1
002000                                                                  L$NAME::
002000  103                                                                     .ASCII /C/
002001  126                                                                     .ASCII /V/
002002  104                                                                     .ASCII /D/
002003  110                                                                     .ASCII /H/
002004  101                                                                     .ASCII /A/
002005  000                                                                     .BYTE   0
```

Program Header

```
002006      000                         .BYTE   0
002007      000                         .BYTE   0
002010                          L$REV::
002010      104                         .ASCII  /D/
002011                          L$DEPO::
002011      060                         .ASCII  /0/
002012                          L$UNIT::
002012  000000                          .WORD   0
002014                          L$TIML::
002014  000016                          .WORD   16
002016                          L$HPCP::
002016  036310                          .WORD   L$HARD
002020                          L$SPCP::
002020  036502                          .WORD   L$SOFT
002022                          L$HPTP::
002022  002214                          .WORD   L$HW
002024                          L$SPTP::
002024  002226                          .WORD   L$SW
002026                          L$LADP::
002026  037020                          .WORD   L$LAST
002030                          L$STA::
002030  000000                          .WORD   0
002032                          L$CO::
002032  000000                          .WORD   0
002034                          L$DTYP::
002034  000000                          .WORD   0
002036                          L$APT::
002036  000000                          .WORD   0
002040                          L$DTP::
002040  002124                          .WORD   L$DISPATCH
002042                          L$PRIO::
002042  000340                          .WORD   PRIO7
002044                          L$ENVI::
002044  000000                          .WORD   0
002046                          L$EXP1::
002046  000000                          .WORD   0
002050                          L$MREV::
002050      004                         .BYTE   C$REVISION
002051      000                         .BYTE   C$EDIT
002052                          L$EF::
002052  000000                          .WORD   0
002054  000000                          .WORD   0
002056                          L$SPC::
002056  000000                          .WORD   0
002060                          L$DEVP::
002060  004036                          .WORD   L$DVTYP
002062                          L$REPP::
002062  024220                          .WORD   L$RPT
002064                          L$EXP4::
002064  000000                          .WORD   0
002066                          L$EXP5::
002066  000000                          .WORD   0
002070                          L$AUT::
002070  000000                          .WORD   0
002072                          L$DUT::
002072  025134                          .WORD   L$DU
002074                          L$LUN::
```

Program Header

```
        002074  000000                          .WORD   0
        002076                          L$DESP::
        002076  004046                          .WORD   L$DESC
        002100                          L$LOAD::
        002100  104035                          EMT     E$LOAD
        002102                          L$ETP::
        002102  003766                          .WORD   L$ERRTBL
        002104                          L$ICP::
        002104  024234                          .WORD   L$INIT
        002106                          L$CCP::
        002106  025100                          .WORD   L$CLEAN
        002110                          L$ACP::
        002110  025076                          .WORD   L$AUTO
        002112                          L$PRT::
        002112  024226                          .WORD   L$PROT
        002114                          L$TEST::
        002114  000001                          .WORD   1
        002116                          L$DLY::
        002116  000000                          .WORD   0
        002120                          L$HIME::
        002120  000000                          .WORD   0
   1156
```

DISPATCH TABLE

```
                                .SBTTL  DISPATCH TABLE

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

                                        DISPATCH 27
002122                                                                  .WORD   27
002124                                                          L$DISPATCH::
002124  025252                                                          .WORD   T1
002126  025542                                                          .WORD   T2
002130  025756                                                          .WORD   T3
002132  026206                                                          .WORD   T4
002134  026370                                                          .WORD   T5
002136  026546                                                          .WORD   T6
002140  026750                                                          .WORD   T7
002142  027142                                                          .WORD   T8
002144  027334                                                          .WORD   T9
002146  027506                                                          .WORD   T10
002150  027706                                                          .WORD   T11
002152  030120                                                          .WORD   T12
002154  030366                                                          .WORD   T13
002156  030730                                                          .WORD   T14
002160  031314                                                          .WORD   T15
002162  031524                                                          .WORD   T16
002164  031740                                                          .WORD   T17
002166  032204                                                          .WORD   T18
002170  032454                                                          .WORD   T19
002172  032562                                                          .WORD   T20
002174  032772                                                          .WORD   T21
002176  033226                                                          .WORD   T22
002200  033526                                                          .WORD   T23
002202  034064                                                          .WORD   T24
002204  035076                                                          .WORD   T25
002206  035750                                                          .WORD   T26
002210  036226                                                          .WORD   T27

```

DISPATCH TABLE

```
                                ;*****************************************************************************
                                ;
                                ;               VDHA.DHT
                                ;
                                ;*****************************************************************************



                                .SBTTL  DEFAULT HARDWARE P-TABLE

                                ;++
                                ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                                ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                                ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
                                ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
                                ;--

002212                                  BGNHW   DFPTBL                                          .WORD   L10000-L$HW/2
002212  000004                                                                          L$HW::
002214                                                                                  DFPTBL::
002214

                                .WORD   160460  ;Default CSR Address
                                .WORD   300     ;Default Vector Address
                                .WORD   177777  ;Default Active lines bit map
                                .BYTE   4       ;Default BR Level
                                .EVEN

                                ENDHW
002224                                                                                  L10000:
```

DEFAULT HARDWARE P-TABLE

```
   1212
   1213                         ;*****************************************************************************
*
   1214                         ;
   1215                         ;               VDHA.SWT
   1216                         ;
   1217                         ;*****************************************************************************
   1218
   1219
   1220
   1221                         .SBTTL  SOFTWARE P TABLE
   1222
   1223                         ;++
   1224                         ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
   1225                         ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
   1226                         ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
   1227                         ; AT RUN TIME.
   1228                         ; -
   1229
   1230 002224                           BGNSW   SFPTBL
        002224  000002                                                    .WORD   L10001-L$SW/2
        002226                                                          L$SW::
        002226                                                          SFPTBL::
   1231
   1232 002226  000021                   OPTION::        .WORD   21      ;bit map of program control flags
   1233 002230  000000                   NDERPT::        .WORD   0       ;Default number of individual data errors to rpt.
   1234
   1235 002232                           ENDSW
        002232                                                          L10001:
```

```
                        ;*****************************************************************************
                        ;
                        ;               VDHA.EQU
                        ;
                        ;*****************************************************************************


                        .SBTTL  GLOBAL EQUATES SECTION




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

000010                          NUMLNS==10      ;NUMBER OF LINES ON DHV11 IS 8.
000377                          MAPLNS==377     ;BIT MAP OF LINES ON DHV11.

                        ;****** DEVICE REGISTER OFFSETS FROM THE CSR'S ADDRESS ******
000000                          CSR0==0         ;CSR REGISTER OFFSET FROM THE CSR ADDRESS
000002                          RBUF0==2        ;RECEIVE REGISTER OFFSET FROM THE CSR ADDRESS
000002                          TXCHR0==2       ;TRANSMIT REGISTER OFFSET FROM THE CSR ADDRESS
000004                          LPR0==4         ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
000006                          STAT0==6        ;STATUS REGISTER OFFSET FROM THE CSR ADDRESS
000010                          LNCTR0==10      ;LINE CONTROL REGISTER OFFSET FROM THE CSR ADDRESS
000012                          TXAD10==12      ;TRANSMIT ADDRESS 1 REGISTER OFFSET FROM THE CSR ADDRESS
000014                          TXAD20==14      ;TRANSMIT ADDRESS 2 REGISTER OFFSET FROM THE CSR ADDRESS
000016                          TXBFC0==16      ;TRANSMIT COUNT REGISTER OFFSET FROM THE CSR ADDRESS

                        ;****** EQUATES USED WITH RESPECT TO THE RX BUFFER ******
000020                          RXBETX==16.     ;LEVEL OF RX BUFFER AT WHICH TO RE-ENABLE TRANSMISSION.
000030                          RXBDTX==24.     ;LEVEL OF RX BUFFER AT WHICH TO DISABLE TRANSMISSION.
000100                          RXBFUL==64.     ;TOTAL CHARACTER CAPACITY OF THE RX BUFFER.


002232                          EQUALS
                        ;
                        ; BIT DIFINITIONS
                        ;
100000                  BIT15== 100000
040000                  BIT14== 40000
020000                  BIT13== 20000
010000                  BIT12== 10000
004000                  BIT11== 4000
002000                  BIT10== 2000
001000                  BIT09== 1000
000400                  BIT08== 400
000200                  BIT07== 200
000100                  BIT06== 100
000040                  BIT05== 40
000020                  BIT04== 20
000010                  BIT03== 10
000004                  BIT02== 4
000002                  BIT01== 2
```

GLOBAL EQUATES SECTION SEQ 0031

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

GLOBAL EQUATES SECTION

```
1301
1302                            ;*****************************************************************************
1303                            ;
1304                            ;               VDHA.GDT
1305                            ;
1306                            ;*****************************************************************************
1307
1308
1309
1310                            .SBTTL  GLOBAL DATA SECTION
1311
1312                            ;++
1313                            ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1314                            ; IN MORE THAN ONE TEST.
1315                            ;--
1316
1317                            ;*****************************************************************************
1318                            ;       Unit Varible Area
1319                            ;*****************************************************************************
1320
1321 002232  000300                      RXVECA:: .WORD  300     ;RX VECTOR ADDRESS.
1322 002234  000304                      TXVECA:: .WORD  304     ;TX VECTOR ADDRESS.
1323 002236  000377                      ACTLNS:: .WORD  377     ;ACTIVE LINE BIT MAP.
1324 002240  000000                      UNITN:: .WORD   0       ;UNIT NUMBER.
1325 002242     004                      BRLEVL:: .BYTE  4       ;INTERRUPT BUS REQUEST LEVEL
1326                                              .EVEN
1327
1328                            ;*****************************************************************************
1329                            ;       Device Register Address Table
1330                            ;*****************************************************************************
1331 002244                              DRADRT::
1332 002244  160020                      CSRA::  .WORD   160020  ;DHV11-M CSR ADDRESS
1333 002246  160022              TXCHA:: RBUFA:: .WORD   160022  ;DHV11-M RECEIVE/TRANSMIT BUFFER ADDRESS
1334 002250  160024                      LPRA::  .WORD   160024  ;DHV11-M LINE PARAMETER REGISTER ADDRESS
1335 002252  160026                      STATA:: .WORD   160026  ;DHV11-M STATUS REGISTER ADDRESS
1336 002254  160030                      LNCTRA:: .WORD  160030  ;DHV11-M LINE CONTROL REGISTER ADDRESS
1337 002256  160032                      TXAD1A:: .WORD  160032  ;DHV11-M TRANSMIT BUFFER 1 REGISTER ADDRESS
1338 002260  160034                      TXAD2A:: .WORD  160034  ;DHV11-M TRANSMIT BUFFER 2 REGISTER ADDRESS
1339 002262  160036                      TXBFCA:: .WORD  160036  ;DHV11-M TRANSMIT BUFFER COUNT REGISTER ADDRESS
1340
1341
1342                            ;*****************************************************************************
1343                            ;       Bit mask table of un-used DHV device register bits.
1344                            ;*****************************************************************************
1345 002264  137660                      UNBTTB::.WORD   137660  ;UNUSED BIT MASK FOR THE CSR
1346 002266  177777                              .WORD   177777  ;UNUSED BIT MASK FOR THE RBUF/TX REG
1347 002270  000007                              .WORD   7       ;UNUSED BIT MASK FOR THE LPR
1348 002272  177777                              .WORD   177777  ;UNUSED BIT MASK FOR THE STAT
1349 002274  166051                              .WORD   166051  ;UNUSED BIT MASK FOR THE LNCTRL
1350 002276  000000                              .WORD   0       ;UNUSED BIT MASK FOR THE TBUFFAD1
1351 002300  077700                              .WORD   77700   ;UNUSED BIT MASK FOR THE TBUFFAD2
1352 002302  000000                              .WORD   0       ;UNUSED BIT MASK FOR THE TBUFFCT
1353
1354                            ;*****************************************************************************
1355                            ;       Register Message Address Table
1356                            ;*****************************************************************************
1357 002304  006004                      RMATBB:: .WORD  DROOMG  ;ADDRESS OF "CSR" MESSAGE.
```

GLOBAL DATA SECTION

```
1358 002306  006010                    .WORD   DR02MG  ;ADDRESS OF "RBUF" MESSAGE.
1359 002310  006015                    .WORD   DR04MG  ;ADDRESS OF "LPR" MESSAGE.
1360 002312  006021                    .WORD   DR06MG  ;ADDRESS OF "STAT" MESSAGE.
1361 002314  006026                    .WORD   DR10MG  ;ADDRESS OF "LNCTRL" MESSAGE.
1362 002316  006035                    .WORD   DR12MG  ;ADDRESS OF "TBUFFAD1" MESSAGE.
1363 002320  006046                    .WORD   DR14MG  ;ADDRESS OF "TBUFAD2" MESSAGE.
1364 002322  006057                    .WORD   DR16MG  ;ADDRESS OF "TBUFFCT" MESSAGE.
1365
1366                            ;*****************************************************************************
1367                            ;       Assorted global variables:
1368                            ;*****************************************************************************
1369 002324  000000                    BUFPTR:: .WORD  0       ;STORAGE FOR RECEIVE CHARACTER BUFFER POINTER.
1370 002326  000001                    TSTNUM:: .WORD  1       ;STORAGE FOR THE TEST NUMBER.
1371 002330  000000                    IESTAT:: .WORD  0       ;STORAGE FOR THE INTERRUPT ENABLE BIT STATES.
1372 002332  000000                    PASCNT:: .WORD  0       ;STO'G FOR PASS COUNT USED IN ROM VERSION# TST.
1373 002334  000000                    RXINTC:: .WORD  0       ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
1374 002336  000000                    RXINTF:: .WORD  0       ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
1375 002340  000000                    TXINTC:: .WORD  0       ;STORAGE FOR TRANSMIT INTERRUPT COUNT.
1376 002342  000000                    TXINTF:: .WORD  0       ;STORAGE FOR TRANSMIT INTERRUPT FLAGS.
1377 002344  000000                    TP4VEC:: .WORD  0       ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
1378 002346  000000                    TP4FLG:: .WORD  0       ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
1379 002350  000000                    WORD1::  .WORD  0       ;LOCATION FOR PASSING INDIRECT PARAMETERS.
1380 002352  000000                    CTRLCF:: .WORD  0       ;STORAGE FOR THE CONTROL-C FLAG.
1381
1382                            ;*****************************************************************************
1383                            ;       Line Time Clock variables and storage.
1384                            ;*****************************************************************************
1385 002354  177546                    CLKCSR:: .WORD  177546  ;CSR ADDRESS OF THE LTC.
1386 002356  000300                    CLKBRL:: .WORD  PRI06   ;INTERRUPT PRIORITY LEVEL OF THE LTC.
1387 002360  000100                    CLKVEC:: .WORD  100     ;INTERRUPT VECTOR ADDRESS OF THE LTC.
1388 002362  000074                    CLKHRZ:: .WORD  60.     ;INTERRUPT FREQUENCY OF THE LTC.
1389 002364  000000                    TIMER1:: .WORD  0       ;HARDWARE CLOCK COUNTER #1.
1390 002366  000000                    TIMER2:: .WORD  0       ;HARDWARE CLOCK COUNTER #2.
1391 002370  000170                    TIMER3:: .WORD  120.    ;HARDWARE BREAK COUNTER LOCATION.
1392 002372  000170                    BCOUNT:: .WORD  120.    ;BREAK COUNT VALUE IN CLOCK TICKS.
1393 002374  000021                    MSTICK:: .WORD  17.     ;NUMBER OF MILLI-SECONDS PER LTC TICK.
1394 002376  000062                    MSLCNT:: .WORD  62      ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.
1395
1396                            ;*****************************************************************************
1397                            ;       Memmory Management Variables and Flags.
1398                            ;*****************************************************************************
1399 002400  177572                    MMSR0:: .WORD   177572  ;ADDRESS OF MEM MGT STATUS REGISTER #0.
1400 002402  000000                    MMPRES:: .WORD  0       ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
1401 002404  000000                    MMENAB:: .WORD  0       ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).
1402 002406  172340                    PAR0A:: .WORD   172340  ;ADDRESS OF MEM MGT PAR #0.
1403
1404                            ;*****************************************************************************
1405                            ;       Table of words with corresponding bit set for generation of bit maps.
1406                            ;*****************************************************************************
1407 002410  000001            BITTBL:: .WORD  1               ;BIT 0 SET.
1408 002412  000002                     .WORD  2               ;BIT 1 SET.
1409 002414  000004                     .WORD  4               ;BIT 2 SET.
1410 002416  000010                     .WORD  10              ;BIT 3 SET.
1411 002420  000020                     .WORD  20              ;BIT 4 SET.
1412 002422  000040                     .WORD  40              ;BIT 5 SET.
1413 002424  000100                     .WORD  100             ;BIT 6 SET.
1414 002426  000200                     .WORD  200             ;BIT 7 SET.
```

GLOBAL DATA SECTION

```
1415 002430  000400                    .WORD   400             ;BIT 8 SET.
1416 002432  001000                    .WORD   1000            ;BIT 9 SET.
1417 002434  002000                    .WORD   2000            ;BIT 10 SET.
1418 002436  004000                    .WORD   4000            ;BIT 11 SET.
1419 002440  010000                    .WORD   10000           ;BIT 12 SET.
1420 002442  020000                    .WORD   20000           ;BIT 13 SET.
1421 002444  040000                    .WORD   40000           ;BIT 14 SET.
1422 002446  100000                    .WORD   100000          ;BIT 15 SET.
1423
1424                            ;*****************************************************************************
1425                            ;*      GPR Save Area Zero.
1426                            ;*****************************************************************************
1427 002450                     GPRS08::                        ;BASE OF GPR SAVE AREA NUMBER ZERO.
1428 002450  000000                    .WORD   0               ;WORD 1, STORAGE FOR R1.
1429 002452  000000                    .WORD   0               ;WORD 2, STORAGE FOR R2.
1430 002454  000000                    .WORD   0               ;WORD 3, STORAGE FOR R3.
1431 002456  000000                    .WORD   0               ;WORD 4, STORAGE FOR R4.
1432 002460  000000                    .WORD   0               ;WORD 5, STORAGE FOR R5.
1433
1434                            ;*****************************************************************************
1435                            ;*      Transmission and Reception Variables, Pointers, and Flags.
1436                            ;*****************************************************************************
1437 002462  000000            ERSMRF:: .WORD  0               ;ERROR SUMMARY REPORT FLAGS.
1438 002464                    ERCNTB:: .BLKW  16.             ;TABLE OF ERROR COUNTERS.
1439
1440                            ;*****************************************************************************
1441                            ;       Storage area for the BMP code queue.
1442                            ;*****************************************************************************
1443 002524  000000            BMPCQP::        .WORD   0       ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
1444 002526                    BMPCQB::        .BLKW   64.     ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
1445 002726                    BMPCQE::                        ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1446                            ;*****************************************************************************
1447                            ;       General table and buffer area--513 words.
14 8                            ;*****************************************************************************
1449 002726                     BUFBAS::                        ;BASE OF MEMORY BUFFER.
1450 002726                     ERLTBL::        .BLKW   128.    ;FIRST HALF OF GENERAL TABLE OR BUFFER.
1451 003326                     BUFMID::        .BLKW   64.     ;SECOND HALF OF GENERAL TABLE OR BUFFER.
1452 003526                     BUF3QT::        .BLKW   64.     ;LAST QUARTER OF THE BUFFER AREA.
1453 003726                     BUFEND::                        ;END OF GENERAL PURPOSE MEMORY BUFFER.
1454 003726                     ENDETB::        .BLKW   16.     ;BUFFER OVERFLOW SPACE.
1455
1468 003766                             ERRTBL
     003766                                                                 L$ERRTBL::
     003766  000000             ERRTYP::        .WORD   0
     003770  000000             ERRNBR::        .WORD   0
     003772  000000             ERRMSG::        .WORD   0
     003774  000000             ERRBLK::        .WORD   0
1469
1470                                    .EVEN
```

GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.

```
1472                            .SBTTL  GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.
1473                            ;*********************************************************************************
1474                            ;*      There are 4 routines and macro definitions used for the handling of
1475                            ;*      GPR values during subroutine calls within this program.  The four
1476                            ;*      routines/macro calls have the following names:
1477                            ;*
1478                            ;*      SAVE - Macro definition used at the beginning of a subroutine to
1479                            ;*              save the GPR contents for later restoration.
1480                            ;*      PASS - Macro definition used at the end of a subroutine to restore
1481                            ;*              the previously saved GPR contents and to leave the contents
1482                            ;*              of the specified GPR(s) intact (NOT restored).
1483                            ;*      PREG05 - Subroutine which is called from the SAVE and PASS macro
1484                            ;*              expansions which actually performs the actions on the GPRs.
1485                            ;*
1486                            ;*      During a subroutine which uses these GPR save routines the values
1487                            ;*      of the GPRs are stored on the stack in the following stack frame:
1488                            ;*
1489                            ;*              SP    -> RET PC INTO PREG05 ROUTINE.
1490                            ;*              SP+2  -> GPR R0 CONTENTS.
1491                            ;*              SP+4  -> GPR R1 CONTENTS.
1492                            ;*              SP+6  -> GPR R2 CONTENTS.
1493                            ;*              SP+8  -> GPR R3 CONTENTS.
1494                            ;*              SP+10 -> GPR R4 CONTENTS.
1495                            ;*              SP+12 -> GPR R5 CONTENTS.
1496                            ;*              SP+14 -> RET PC INTO CALLER OF SUB'TNE WHICH CALLED PREG05.
1497                            ;*
1498                            ;*      Each level of sub'tne calling uses 8 words of stack overhead.
1499                            ;*      The SAVE and PASS macros can also be used in "straight line code"
1500                            ;*      to save and restore the GPR values.  In any case, after the
1501                            ;*      issuing of a PASS call the GPRs will be restored to the values
1502                            ;*      they had prior to the last SAVE call (except for the excepted,
1503                            ;*      or passed intact, GPRs specified as parameters to the PASS call)
1504                            ;*      and the SP will also be restored to its condition before the last
1505                            ;*      SAVE call.  The programmer must be sure that the SP has the same
1506                            ;*      value when the PASS macro is called as it had immediately after
1507                            ;*      the SAVE macro was called.
1508                            ;*********************************************************************************
```

GPR FRAME ACCESS EQUATES SEQ 0036

```
1510                                .SBTTL GPR FRAME ACCESS EQUATES
1511                                ;+++
1512                                ;Equates that allow access to the stack frame.  These are the
1513                                ;offsets into the stack for registers saved during the PREG05
1514                                ;routine.
1515                                ;---
1516
1517        000036                  LPCSLT==        36      ;Offset for last return PC.
1518        000016                  PCSLOT==        16      ;Offset for return PC.
1519        000014                  R5SLOT==        14      ;Offset for R5.
1520        000012                  R4SLOT==        12      ;Offset for R4.
1521        000010                  R3SLOT==        10      ;Offset for R3.
1522        000006                  R2SLOT==        6       ;Offset for R2.
1523        000004                  R1SLOT==        4       ;Offset for R1.
1524        000002                  R0SLOT==        2       ;Offset for R0.
```

GLOBAL MACRO DEFINITION - SAVE -

```
1526                            .SBTTL  GLOBAL MACRO DEFINITION         - SAVE -
1527                            ;*******************************************************************************
1528                            ;*      This macro is used at the beginning of a subroutine to save the
1529                            ;*      contents of the GPRs R0 thru R5.
1530                            ;*
1531                            ;* INPUTS:      SP - Unchanged since subroutine was entered
1532                            ;*              R5SLOT - Offset to stack slot for R5 (Equated to 14 Octal)
1533                            ;*
1534                            ;* OUTPUTS:     GPR save area on the stack is loaded with the contents of GPRs
1535                            ;*              TOP OF STACK - Loaded with the return address into PREG05
1536                            ;*
1537                            ;* CALLING SEQUENCE:    SAVE
1538                            ;*
1539                            ;* COMMENTS:    No arguments are allowed.
1540                            ;*              The PASS macro should be called to restore the GPR values.
1541                            ;*
1542                            ;* SUBORDINATE ROUTINES CALLED: PREG05.
1543                            ;*******************************************************************************
1544
1545                                    .MACRO  SAVE
1546                                    .LIST
1547                                            JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
1548                                    .NLIST
1549                                    .ENDM   SAVE
```

```
1551                                    .SBTTL  GLOBAL MACRO DEFINITION          - PASS -
1552                                    ;*************************************************************************
1553                                    ;*      This macro is used in conjunction with the SAVE macro.  It is
1554                                    ;*      called at end of a subroutine to pass parameters in GPRs back to the
1555                                    ;*      calling routine by altering the GPR save area on the stack and then
1556                                    ;*      returning to PREG05 to restore the GPRs to their saved values.
1557                                    ;*
1558                                    ;* INPUTS:      Only allowed ARGUMENTS are "R0" thru "R5".
1559                                    ;*              R0SLOT thru R5SLOT must be equated to their respective GPR save
1560                                    ;*                      slot offsets before calling this macro.
1561                                    ;*
1562                                    ;* OUTPUTS:     The GPR values are put in their respective slots on the stack.
1563                                    ;*
1564                                    ;* CALLING SEQUENCE:    PASS    R0,R1,...
1565                                    ;*
1566                                    ;* COMMENTS:    Any combination of GPR arguments may be listed in any order.
1567                                    ;*              For example, the following are legal:
1568                                    ;*                      PASS    R1
1569                                    ;*                      PASS    R4,R0,R2
1570                                    ;*              The GPRs listed as arguments will be passed intact to the
1571                                    ;*              calling routine, all other GPRs will be restored.
1572                                    ;*              The SP must be at its original value when PASS is called.
1573                                    ;*
1574                                    ;*              The macro call
1575                                    ;*                      PASS    R0,R3
1576                                    ;*              expands into the following assembly code:
1577                                    ;*                      MOV     R0,R0SLOT(SP)           ;PUT R0 IN STACK SLOT.
1578                                    ;*                      MOV     R3,R3SLOT(SP)           ;PUT R3 IN STACK SLOT.
1579                                    ;*                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
1580                                    ;*              In this example GPRs R1, R2, R4, and R5 will be restored to
1581                                    ;*              their values contained in the stack frame and R0 and R3
1582                                    ;*              will be left at their values prior to this PASS call.
1583                                    ;*
1584                                    ;* SUBORDINATE ROUTINES CALLED: (PREGRT - Label within PREG05, value on stack.)
1585                                    ;*************************************************************************
1586
1587                                            .MACRO  PASS    A,B,C,D,E,F
1588                                            .IRP    X,<A,B,C,D,E,F>
1589                                            .IF     NB,X
1590                                            .LIST
1591                                                    MOV     X,X'SLOT(SP)            ;PUT X IN STACK SLOT.
1592                                            .NLIST
1593                                            .ENDC
1594                                            .ENDM
1595                                            .LIST
1596                                                    JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
1597                                            .NLIST
1598                                            .ENDM   PASS
```

GLOBAL SUBROUTINE - PREG05 -

```
                                .SBTTL GLOBAL SUBROUTINE                - PREG05 -
                                ;*******************************************************************************
                                ;*      Preserve Registers R0 through R5 for subroutine calls.
                                ;*
                                ;* INPUTS:      The return address back into the calling routine must be in
                                ;*              GPR R5.  (i.e.- Macros use "JSR  R5,PREG05".)
                                ;*
                                ;* OUTPUTS:     Registers R0 through R5 are saved on the stack.
                                ;*
                                ;*CALLING SEQUENCE:     SAVE            ;Macro expansion calls PREG05.
                                ;*                      [Subroutine code]...
                                ;*                      PASS            ;Macro expansion recalls PREG05.
                                ;*
                                ;*COMMENTS:     This routine is re-entrant.
                                ;*
                                ;*              Parameters may be passed out of a subroutine by modifying the
                                ;*              register save area on the stack.  Use the PASS GPRn macro
                                ;*              to return GPR values intact.
                                ;*              Use the RnSLOT offsets from the SP to pass other parameters.
                                ;*              [Example:       MOV    VALUE,R0SLOT(SP)        ]
                                ;*              Make sure the SP is at its original value when you do this.
                                ;*
                                ;*SUBORDINATE ROUTINES CALLED:  None.
                                ;*******************************************************************************

003776                          PREG05:                 ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
003776  010446                          MOV     R4,-(SP)        ;SAVE R4
004000  010346                          MOV     R3,-(SP)        ;SAVE R3
004002  010246                          MOV     R2,-(SP)        ;SAVE R2
004004  010146                          MOV     R1,-(SP)        ;SAVE R1
004006  010046                          MOV     R0,-(SP)        ;SAVE R0
004010  010546                          MOV     R5,-(SP)        ;PUSH RETURN PC ON TOP OF STACK
004012  016605  000014                  MOV     R5SLOT(SP),R5   ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS

004016  004736                          JSR     PC,@(SP)+       ;Call the subroutine at the return address
                                                                ;from the PREG05 call, putting the present
                                                                ;PC on the stack as a return address into
                                                                ;this (PREG05) routine.

                                        ;+++
                                        ;The following code is executed when the calling routine does a
                                        ;"return" [JSR  PC,@(SP)+] using the PC deposited on the stack above.
                                        ;---

004020  012605                  PREGRT:: MOV    (SP)+,R5        ;Put return PC in R5.
004022  012600                          MOV     (SP)+,R0        ;Restore R0.
004024  012601                          MOV     (SP)+,R1        ;Restore R1.
004026  012602                          MOV     (SP)+,R2        ;Restore R2.
004030  012603                          MOV     (SP)+,R3        ;Restore R3.
004032  012604                          MOV     (SP)+,R4        ;Restore R4.

004034  000205                          RTS     R5              ;Return to the subroutine which called PREG05.
                                                                  ;restoring R5 in the process.
```

GLOBAL TEXT SECTION

```
                        .SBTTL  GLOBAL TEXT SECTION
                        ;**********************************************************************
                        ;
                        ;               FVTSKL1.P11
                        ;
                        ;**********************************************************************


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

                        ;
                        ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                        ;
004036                          DEVTYP  <DHV11-M>
004036                                                          L$DVTYP::
004036  104     110     126                                             .ASCIZ  %DHV11-M%
004041  061     061     055
004044  115     000
                                                                        .EVEN


                        ; TEST DESCRIPTION
                        ;
004046                          DESCRIPT        <DHV11 M FUNC TST PART 1>
004046                                                          L$DESC::
004046  104     110     126                                             .ASCIZ  /DHV11-M FUNC TST PA
RT 1/
004051  061     061     055
004054  115     040     106
004057  125     116     103
004062  040     124     123
004065  124     040     120
004070  101     122     124
004073  040     061     000
                                                                        .EVEN
                        .EVEN
```

```
                                        ;*******************************************************************************************
*
                                        ;
                                        ;               VDHA.FMT
                                        ;
                                        ;*************************************************************************************



                                        ;
                                        ; FORMAT STATEMENTS USED IN PRINT CALLS
                                        ;
```

```
                                    ;*********************************************************************
                                    ;
                                    ;               VDHA.MSG
                                    ;
                                    ;*********************************************************************
**


            .NLIST BIN
            .SBTTL  GLOBAL MESSAGE AREA
                    ; ********* FORMAT STATEMENTS *********
004076  MFUNIT:: .ASCIZ /*N*A TESTING UNIT :*D4*A(D)*N/
004134  EF0503:: .ASCIZ /*T*N/
004141  EF0505:: .ASCIZ /*A    *D5*A ILLEGAL INTERRUPTS RECEIVED.*N/
004214  EF1401:: .ASCIZ /*N*A ROM VERSION NUMBERS: PROC_1 = *D2*A(D)   PROC_2 = *D2*A(D)*N/
004316  EF1402:: .ASCIZ /*T*A ROM VERSION NUMBER *T*N/
004353  EF1601:: .ASCIZ /*A  *T*A ABORTED *N/
004377  EF1602:: .ASCIZ /*A    EXPECTED DATA: *O6*A (O).*N/
004441  EF1603:: .ASCIZ /*A    ACTUAL DATA:   *O6*A (O).*N/
004503  EF1604:: .ASCIZ /*A  BAD BIT(S) IN DEVICE *T*A REGISTER FOR LINE *D2*A (D).*N/
004600  EF3001:: .ASCIZ /*A    EXPECTED OR CORRECT VALUE: *O3*N/
004647  EF3002:: .ASCIZ /*A    ACTUAL OR MEASURED  VALUE: *O3*N/
004716  EF9001:: .ASCIZ /*A  UNEXPECTED *T*A FOUND IN RECEIVE CHAR FIFO:*N/
005000  EF9002:: .ASCIZ /*A    CODE IS ASSOCIATED WITH LINE: *D2*A(D)*N/
005057  EF9003:: .ASCIZ /*A      CODE IS: *O3*A(O)*N/
005113  EF9004:: .ASCIZ /*A    *T*A VALUE: *O3*A(O)*N/
005150  EF9005:: .ASCIZ /*A    *T*A VALUE: NONE*N/
005201  EF9006:: .ASCIZ /*A  *T*A *D2*A(D)*N/
005225  EF9010:: .ASCIZ /*A    NUMBER OF ERRORS DETECTED ON LINE *D2*A(D) IS *D5*A(D)*N/
005324  EF9016:: .ASCIZ /*A  UNEXPECTED *T*A FOR LINE *D2*A(D) IN FIFO AFTER RESET:*N/
005421  EF9017:: .ASCIZ /*A    *T*A (WITH ERROR FLAGS) IS *O6*A(O)*N/
005475  EF9018:: .ASCII /*A  *T*A IN SELFTEST CODE FIFO SLOT FOR LINE *D2/
005555           .ASCIZ /*A(D) AFTER RESET.*N/
005602  EF9019:: .ASCIZ /*A  *T*A *O6*A(O)*N/
005626  EF9301:: .ASCIZ /*A *T*D2*A(D),  BMP CODE REPORTED :*O3*A(O)*N/
005704  EF9302:: .ASCIZ /*A  OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)*N/
                 ;********* ERROR MESSAGES ***********
006004  DR00MG:: .ASCIZ /CSR/
006010  DR02MG:: .ASCIZ /RBUF/
006015  DR04MG:: .ASCIZ /LPR/
006021  DR06MG:: .ASCIZ /STAT/
006026  DR10MG:: .ASCIZ /LNCTRL/
006035  DR12MG:: .ASCIZ /TBUFFAD1/
006046  DR14MG:: .ASCIZ /TBUFFAD2/
006057  DR16MG:: .ASCIZ /TBUFFCT/
006067  EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
006125  EM0201:: .ASCIZ /MASTER RESET (PERFORM SELFTEST) TEST /
006173  EM0202:: .ASCIZ /  MASTER RESET BIT DID NOT CLEAR AFTER BOARD RESET./
006257           .ASCIZ /    WAITED 5 SECONDS.  BIT DEFECTIVE OR FIRMWARE HUNG./
006346  EM0203:: .ASCIZ /  MASTER RESET BIT CLEAR IMMEDIATELY AFTER BOARD RESET./
006436           .ASCIZ /    BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
006511  EM0204:: .ASCIZ \  MR BIT WENT CLEAR WITHIN 1/2 SECOND OF BOARD RESET.\
006577           .ASCIZ /    BIT DEFECTIVE OR SELFTEST WAS (INCORRECTLY) SKIPPED./
006670  EM0301:: .ASCIZ /MASTER RESET (SKIP SELFTEST) TEST /
006733  EM0302:: .ASCIZ /  MR BIT CLR WITHIN 10 MILISECOND AFTER BOARD RESET./
007020           .ASCIZ /    BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
007073  EM0303:: .ASCIZ \  MR BIT WENT CLEAR 1/5 TO 5 SECONDS AFTER RESET.\
```

GLOBAL MESSAGE AREA

```
1784 007155            .ASCIZ /    SELFTEST DID NOT GET SKIPPED (SHOULD HAVE BEEN SKIPPED)./
1785 007252  EM0401:: .ASCIZ /RBUF REGISTER RX CHARACTER FIELD TEST /
1786 007321  EM0402:: .ASCIZ /  IMPROPER CODE FOUND IN RX FIFO AFTER DUT RESET./
1787 007403            .ASCIZ /    EXPECTED: SELFTEST CODE,   ACTUAL: IMPROPER CODE./
1788 007471  EM0501:: .ASCIZ /RBUF REGISTER ERROR FLAGS FIELD TEST /
1789 007537  EM0502:: .ASCIZ /  RX ERROR FLAG(S) FOUND CLEAR ON SELFTEST CODE./
1790 007620            .ASCIZ /    EXPECTED: ALL ERROR FLAGS SET,  ACTUAL: FLAG(S) CLEAR./
1791 007713  EM0525:: .ASCIZ /  RX INTERRUPT(S) RECEIVED WITH RX INTERRUPTS DISABLED./
1792 010003  EM0526:: .ASCIZ /  TX INTERRUPT(S) RECEIVED WITH TX INTERRUPTS DISABLED./
1793 010073  EM0601:: .ASCIZ /CSR RX.DATA.AVAIL BIT TEST /
1794 010127  EM0602:: .ASCIZ /  RX.DATA.AVAIL BIT FOUND CLEAR AFTER RESFT COMPLETION./
1795 010217            .ASCIZ /    EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
1796 010307  EM0603:: .ASCIZ /  RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO./
1797 010401            .ASCIZ /    600 CHARS READ FROM FIFO WITHOUT R.D.A BIT CLEARING./
1798 010472  EM0701:: .ASCIZ /RBUF RX.DATA.VALID BIT TEST /
1799 010527  EM0702:: .ASCIZ /  RX.DATA.VALID BIT FOUND CLEAR AFTER RESET COMPLETION./
1800 010617            .ASCIZ /    EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
1801 010707  EM0703:: .ASCIZ /  RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO./
1802 011001            .ASCIZ /    600 CHARS READ FROM FIFO WITHOUT R.D.V BIT CLEARING./
1803 011072  EM0801:: .ASCIZ /RBUF RX.LINE.NUMBER FIELD TEST /
1804 011132  EM0802:: .ASCIZ /  LINE NUMBER IS WRONG ON A SELFTEST CODE./
1805 011205  EM0901:: .ASCIZ /CHECK FOR BMP_CODES TEST/
1806 011236  EM0902:: .ASCIZ /UNEXPECTED BMP CODES FOUND./
1807 011272  EM1001:: .ASCIZ /DIAGNOSTIC FAIL (SKP SELFTEST) TEST/
1808 011336  EM1002:: .ASCIZ /  SKIP SELF TEST TOOK TOO LONG TO COMPLETE, > 50 MS./
1809 011423  EM1003:: .ASCIZ /  SKIP SELF-TEST COMPLETED TOO SOON, < 10 MS./
1810 011501  EM1101:: .ASCIZ /SKIP SELF-TEST TEST/
1811 011525  EM1201:: .ASCIZ /SELF-TEST TEST/
1812 011544  EM1202:: .ASCIZ /  SELF-TEST TOOK TOO LONG TO COMPLETE, > 3 SECONDS./
1813 011630  EM1203:: .ASCIZ \  SEL- TEST COMPLETED TOO SOON, < 1/2 SECOND.\
1814 011706  EM1204:: .ASCIZ /  SELF-TEST DID NOT EXECUTE/
1815 011742  EM1205:: .ASCIZ /  DIAG_FAIL BIT BAD/
1816 011766  EM1301:: .ASCIZ /FAIL SELF-TEST TEST/
1817 012012  EM1302:: .ASCIZ / SELF-TEST ERROR REPORTING BAD/
1818 012051  EM1401:: .ASCIZ /ROM VERSION_NUMBER TEST/
1819 012101  EM1402:: .ASCIZ /  FIFO EMPTY, ONE OR MORE ROM VERSION_NUMBERS MISSING/
1820 012167  EM1403:: .ASCIZ /  ROM VERSION_NUMBER FOUND OUT OF SEQUENCE/
1821 012242  EM1404:: .ASCIZ /  ONE OR MORE ROM VERSION_NUMBERS MISSING/
1822 012314  EM1405:: .ASCIZ /    PROC_1/
1823 012327  EM1406:: .ASCIZ /    PROC_2/
1824 012342  EM1407:: .ASCIZ /NOT FOUND/
1825 012354  EM1408:: .ASCIZ /FOUND/
1826 012362  EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
1827 012445  EM1604:: .ASCIZ \DEVICE REGISTER WORD READ/WRITE TEST \
1828 012513  EM1701:: .ASCIZ \DEVICE REGISTER WORD READ/MODIFY/WRITE TEST \
1829 012570  EM1801:: .ASCIZ \DEVICE REGISTER BYTE READ/WRITE TEST \
1830 012636  EM1901:: .ASCIZ \DEVICE REGISTER BYTE READ/MODIFY/WRITE TEST \
1831 012713  EM2001:: .ASCIZ /DEVICE STAT REGISTER ID BIT TEST /
1832 012755  EM2002:: .ASCIZ /ID BIT BAD.  EXPECTED: CLEAR, ACTUAL: SET./
1833 013030  EM2101:: .ASCIZ \NO TX_DATA_VALID/NO TX_ACTION TEST\
1834 013073  EM2102:: .ASCIZ /  TX_ACTION FOUND AFTER INVALID DATA WORD WRITTEN TO LINE: /
1835 013167  EM2201:: .ASCIZ \TX_DATA_VALID/TX_ACTION TEST\
1836 013224  EM2202:: .ASCIZ /  NO TX_ACTION FOUND AFTER VALID DATA WORD TX'D ON LINE: /
1837 013316  EM2203:: .ASCIZ /  INCORRECT LINE NUMBER FOUND WITH TX_ACT AFTER DATA TX'D ON LINE : /
1838 013423  EM230':: .ASCIZ /TX_ENABLE (INACTIVE) BIT TEST/
1839 013461  EM23C2:: .ASCIZ /  TX_ENABLE BIT BAD ON LINE. /
1840 013517  EM2?01:: .ASCIZ /TX_ENABLE (ACTIVE) BIT TEST/
```

```
1841 013553  EM2601:: .ASCIZ /RECEIVE INTERRUPT TEST /
1842 013603  EM2602:: .ASCIZ /  NO RX INT GENERATED (DATA_VALID SET, RX INTS ENABLED)./
1843 013674  EM2603:: .ASCIZ /  NO RX INT GENERATED (NO CODES IN FIFO AFTER RESET)./
1844 013762  EM2604:: .ASCIZ /  NO RX INT GENERATED (RX_DATA_AVAIL CLR, RX INTS ENABLED)./
1845 014056  EM2605:: .ASCIZ /  RX INTERRUPT GENERATED WITH RX_DATA_AVAIL CLEAR./
1846 014141  EM2606:: .ASCIZ /TRANSMIT INTERRUPT TEST ERROR:/
1847 014200  EM2607:: .ASCIZ /  TX_ACTION SET REPEATEDLY AFTER BOARD RESET, NO DATA SENT./
1848 014274  EM2608:: .ASCIZ /  TX_ACTION STUCK SET AFTER BOARD RESET./
1849 014345  EM2609:: .ASCIZ /  TX INTERRUPT GENERATED WITH TX_ACTION CLEAR./
1850 014424  EM2610:: .ASCIZ /  NO TX INTERRUPT WITH TX_ACTION SET AND TX INTS ENABLED./
1851 014516  EM2611:: .ASCIZ /  TX_ACTION NOT SET AFTER CHARS SENT ON ALL LINES./
1852 014601  EM3001:: .ASCIZ /INTERRUPT BR LEVEL TEST /
1853 014632  EM3002:: .ASCIZ /  NO RX_DATA_AVAIL FROM SELFTEST CODES IN FIFO AFTER RESET./
1854 014726  EM3003:: .ASCIZ /  TX INTERRUPT GENERATED AT WRONG BR LEVEL:/
1855 015002  EM3004:: .ASCIZ /  RX INTERRUPT GENERATED AT WRONG BR LEVEL:/
1856 015056  EM3005:: .ASCIZ /  TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS RX INT./
1857 015150  EM3101:: .ASCIZ /DIAGNOSTIC FIELD (BMP) TEST/
1858 015204  EM3102:: .ASCIZ /  DIAGNOSTIC FIELD BAD ON LINE: /
1859 015245  EM9009:: .ASCIZ /EXPECTED OR CORRECT/
1860 015271  EM9010:: .ASCIZ /ACTUAL OR MEASURED /
1861 015315  EM9014:: .ASCIZ /SUMMARY REPORTS FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:/
1862 015411  EM9017:: .ASCII /  FIFO WILL NOT PURGE (DATA.VALID STUCK SET),/
1863 015466           .ASCIZ / REMAINDER OF TEST SKIPPED./
1864 015522  EM9018:: .ASCIZ /NO CODE/
1865 015532  EM9019:: .ASCIZ /NON-SELFTEST/
1866 015547  EM9020:: .ASCIZ /SELFTEST ERROR CODE/
1867 015573  EM9022:: .ASCIZ /DATA CHARACTER/
1868 015612  EM9023:: .ASCIZ /MODEM STATUS CODE/
1869 015634  EM9024:: .ASCIZ /SELFTEST CODE/
1870 015652  EM9026:: .ASCIZ /     LPR CONTENTS: /
1871 015676  EM9301:: .ASCIZ /BMP CODE REPORT/
1872 015716  EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
1873 015746  EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
1874 016013  EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
1875                  .EVEN
1876                                .LIST BIN
```

```
1878
1879
1880                    ;*****************************************************************************************************
1881                    ;
1882                    ;               FVTSKL2.P11
1883                    ;
1884                    ;*****************************************************************************************************
1885
1886
1887
1888                    .SBTTL  GLOBAL ERROR REPORT SECTION
1889
1890                    ;++
1891                    ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
1892                    ; USED BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
1893                    ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
1894                    ;--
```

```
 1896                                   .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0101 -
 1897                                   ;*************************************************************************
 1898                                   ;*      This is an error reporting subroutine which prints additional error
 1899                                   ;*      information if an error is detected in TEST 1 (Register Address
 1900                                   ;*      Access Test).  This subroutine reports the type of access (Read or
 1901                                   ;*      Write or both) which caused a bus time-out trap (004 trap).
 1902                                   ;*      A message indicating that the DHV may be at the wrong Q-bus address
 1903                                   ;*      is also printed.
 1904                                   ;*
 1905                                   ;* INPUTS:      R5 - Error flag word.
 1906                                   ;*                      If bit 0 is set, a read error occured.
 1907                                   ;*                      If bit 1 is set, a write error occured.
 1908                                   ;*
 1909                                   ;* OUTPUTS:     Messages are printed at the operator console.
 1910                                   ;*
 1911                                   ;* CALLING SEQUENCE:    Include the label "ER0101" as the message pointer
 1912                                   ;*                      parameter in the DRS error report macro call.
 1913                                   ;*
 1914                                   ;* COMMENTS:
 1915                                   ;*
 1916                                   ;* SUBORDINATE ROUTINES USED: None.
 1917                                   ;*************************************************************************
 1918
 1919 016070                                    BGNMSG  ER0101
      016070
 1920 016070                                    SAVE                                    ER0101::
      016070  004537  003776                            JSR     R5,PREG05       ;SAVE THE GPR CONTENTS.
 1921                                                                           ;CALL REGISTER SAVE SUBRT.
 1922 016074  032705  000001                    BIT     #BIT0,R5        ;TEST FOR READ ERROR.
 1923 016100  001410                            BEQ     2$              ;SKIP READ ERROR MSG IF NO READ ERROR.
 1924 016102                                    PRINTB  #MSG1           ;PRINT READ ERROR MESSAGE.
      016102  012746  016174                                                    MOV     #MSG1,-(SP)
      016106  012746  000001                                                    MOV     #1,-(SP)
      016112  010600                                                            MOV     SP,R0
      016114  104414                                                            TRAP    C$PNTB
      016116  062706  000004                                                    ADD     #4,SP
 1925 016122  032705  000002            2$:     BIT     #BIT1,R5        ;TEST FOR WRITE ERROR.
 1926 016126  001410                            BEQ     4$              ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
 1927 016130                                    PRINTB  #MSG2           ;PRINT WRITE ERROR MESSAGE.
      016130  012746  016252                                                    MOV     #MSG2,-(SP)
      016134  012746  000001                                                    MOV     #1,-(SP)
      016140  010600                                                            MOV     SP,R0
      016142  104414                                                            TRAP    C$PNTB
      016144  062706  000004                                                    ADD     #4,SP
 1928 016150                            4$:     PRINTX  #MSG3           ;SUGGEST THAT DHV MAY BE AT WRONG ADDRESS.
      016150  012746  016331                                                    MOV     #MSG3,-(SP)
      016154  012746  000001                                                    MOV     #1,-(SP)
      016160  010600                                                            MOV     SP,R0
      016162  104415                                                            TRAP    C$PNTX
      016164  062706  000004                                                    ADD     #4,SP
 1929 016170                                    PASS                    ;RESTORE THE GPR CONTENTS.
      016170  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
 1930 016172                                    ENDMSG
      016172                                                            L10002:
      016172  104423                                                            TRAP    C$MSG
 1931
 1932 016174     045     101     102    MSG1::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.%N/
```

```
        016177  125  123  040
        016202  124  111  115
        016205  105  055  117
        016210  125  124  040
        016213  124  122  101
        016216  120  040  103
        016221  101  125  123
        016224  105  104  040
        016227  102  131  040
        016232  122  105  101
        016235  104  040  101
        016240  124  124  105
        016243  115  120  124
        016246  056  045  116
        016251  000
1933    016252  045  101  102  MSG2::   .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.%N/
        016255  125  123  040
        016260  124  111  115
        016263  105  055  117
        016266  125  124  040
        016271  124  122  101
        016274  120  040  103
        016277  101  125  123
        016302  105  104  040
        016305  102  131  040
        016310  127  122  111
        016313  124  105  040
        016316  101  124  124
        016321  105  115  120
        016324  124  056  045
        016327  116  000
1934    016331  045  101  104  MSG3::   .ASCIZ  /%ADHV MAY BE AT THE WRONG Q-BUS ADDRESS.%N%N/
        016334  110  126  040
        016337  115  101  131
        016342  040  102  105
        016345  040  101  124
        016350  040  124  110
        016353  105  040  127
        016356  122  117  116
        016361  107  040  121
        016364  055  102  125
        016367  123  040  101
        016372  104  104  122
        016375  105  123  123
        016400  056  045  116
        016403  045  116  000
1935
1936                                .EVEN
```

GLOBAL ERROR REPORTING ROUTINE - ER0201 -

```
                                .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0201 -
                                ;**********************************************************************
                                ;*      This is an error reporting subroutine which prints 2 contiguous
                                ;*      ASCII error messages.  The address of the first message is passed
                                ;*      as an input parameter and the address of the second is found by
                                ;*      searching for the end of the first message.
                                ;*
                                ;* INPUTS:      R1 - Address of the first message to print.
                                ;*
                                ;* OUTPUTS:     A messages is printed at the operator console.
                                ;*
                                ;* CALLING SEQUENCE:    Load the address of the first message in R1.
                                ;*                      Include the label "ER0201" as the message pointer
                                ;*                      parameter in the Diag Super error report macro call.
                                ;*
                                ;* COMMENTS:    The message is printed as Basic error information.
                                ;*              The second message should follow the first one in the program
                                ;*              memory.  Each message should be defined using .ASCIZ
                                ;*
                                ;* SUBORDINATE ROUTINES USED: None.
                                ;**********************************************************************

016406                                  BGNMSG  ER0201
016406                                                                          ER0201::
016406                                  SAVE                    ;SAVE THE GPR CONTENTS.
016406  004537  003776                                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

016412  010102                           MOV    R1,R2
016414  105722                  2$:      TSTB   (R2)+           ;CHECK FOR A ZERO BYTE (END OF MESSAGE).
016416  001376                          BNE    2$              ;LOOP UNTIL NEXT MESSAGE IS FOUND.

016420                                  PRINTB  #EF0503,R1      ;PRINT THE FIRST MESSAGE.
016420  010146                                                                  MOV     R1,-(SP)
016422  012746  004134                                                         MOV     #EF0503,-(SP)
016426  012746  000002                                                         MOV     #2,-(SP)
016432  010600                                                                 MOV     SP,R0
016434  104414                                                                 TRAP    C$PNTB
016436  062706  000006                                                         ADD     #6,SP
016442                                  PRINTB  #EF0503,R2      ;PRINT THE SECOND MESSAGE.
016442  010246                                                                 MOV     R2,-(SP)
016444  012746  004134                                                         MOV     #EF0503,-(SP)
016450  012746  000002                                                         MOV     #2,-(SP)
016454  010600                                                                 MOV     SP,R0
016456  104414                                                                 TRAP    C$PNTB
016460  062706  000006                                                         ADD     #6,SP

016464                                  PASS                    ;RESTORE THE GPR CONTENTS.
016464  004736                                        JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.

016466                                  ENDMSG
016466                                                                  L10003:
016466  104423                                                                 TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0503 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which prints an additional error
                        ;*      message whose address is passed as an input parameter.
                        ;*
                        ;* INPUTS:      R1 - Address of the message to print.
                        ;*
                        ;* OUTPUTS:     A messages is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Load the address of the message in R1.
                        ;*                      Include the label "ER0503" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The message is printed as Basic error information.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*************************************************************************

016470                          BGNMSG  ER0503
016470                                                                  ER0503::
                        
016470                          PRINTB  #EF0503,R1      ;PRINT THE MESSAGE.
016470  010146                                                              MOV     R1,-(SP)
016472  012746  004134                                                      MOV     #EF0503,-(SP)
016476  012746  000002                                                      MOV     #2,-(SP)
016502  010600                                                              MOV     SP,R0
016504  104414                                                              TRAP    C$PNTB
016506  062706  000006                                                      ADD     #6,SP
                        
016512                          ENDMSG
016512                                                          L10004:
016512  104423                                                              TRAP    C$MSG
```

GLOBAL ERROR REPORTING ROUTINE - ER0504 -

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0504 -
                        ;*********************************************************************
                        ;*      This is an error reporting subroutine which prints additional error
                        ;*      messages when illegal interrupts are received.
                        ;*
                        ;* INPUTS:      R1 - Address of the message to print.
                        ;*              R2 - Number of illegal interrupts received.
                        ;*
                        ;* OUTPUTS:     Messagess are printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Load the address of the message in R1.
                        ;*                      Load the number of illegal ints in R2.
                        ;*                      Include the label "ER0504" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*********************************************************************

016514                          BGNMSG  ER0504
016514                                                          ER0504::

                        PRINTB  #EF0503,R1      ;PRINT THE FIRST LINE OF THE MESSAGE.
016514  010146                                                 MOV     R1,-(SP)
016516  012746  004134                                        MOV     #EF0503,-(SP)
016522  012746  000002                                        MOV     #2,-(SP)
016526  010600                                                 MOV     SP,R0
016530  104414                                                 TRAP    C$PNTB
016532  062706  000006                                        ADD     #6,SP
                        PRINTX  #EF0505,R2      ;PRINT THE NUMBER OF INTS RECEIVED.
016536  010246                                                 MOV     R2,-(SP)
016540  012746  004141                                        MOV     #EF0505,-(SP)
016544  012746  000002                                        MOV     #2,-(SP)
016550  010600                                                 MOV     SP,R0
016552  104415                                                 TRAP    C$PNTX
016554  062706  000006                                        ADD     #6,SP

016560                          ENDMSG
016560                                                  L10005:
016560  104423                                                 TRAP    C$MSG
```

GLOBAL ERROR REPORTING ROUTINE - ER1401 -

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER1401 -
                        ;*****************************************************************************
                        ;*      This is an error reporting subroutine which prints additional error
                        ;*      information if an error is detected in the Rom version test.
                        ;*      This subroutine analyses the input parameters which contain the
                        ;*      ROM version numbers for PROC_1 and PROC_2 and reports the appropriate
                        ;*      error message to the operator.
                        ;*
                        ;* INPUTS:      R1 - Contains the address of the first message to be reported.
                        ;*              R3 - Contains the ROM version number of PROC_1.
                        ;*              R4 _ Contains the ROM version number of PROC_2.
                        ;*
                        ;* OUTPUTS:     Basic and extended error messages are reported at the
                        ;*              operators console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER1401" as the message pointer
                        ;*                      parameter in the DRS error report macro call.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*****************************************************************************

016562                          BGNMSG  ER1401
016562                                                                  ER1401::

016562                          PRINTB  #EF0503,R1              ;REPORT THE ERROR MESSAGE PASSED IN.
016562  010146                                                          MOV     R1,-(SP)
016564  012746  004134                                                  MOV     #EF0503,-(SP)
016570  012746  000002                                                  MOV     #2,-(SP)
016574  010600                                                          MOV     SP,R0
016576  104414                                                          TRAP    C$PNTB
016600  062706  000006                                                  ADD     #6,SP

                        ;+
                        ; Determine which Rom version number(s) are missing.
                        ;-

016604  012705  000143          MOV     #99.,R5                 ;GET INVALID ROM NUMBER.
016610  012701  012314          MOV     #EM1405,R1              ;SELECT PROC_1 MESSAGE.
016614  012702  012342          MOV     #EM1407,R2              ;SELECT THE "NOT FOUND" MESSAGE.
016620  120305                  CMPB    R3,R5                   ;CHECK PROC_1 ROM VERSION NUMBER.
016622  001402                  BEQ     2$                      ;GO REPORT PROC_1 CODE NOT FOUND.
016624  012702  012354          MOV     #EM1408,R2              ;SELECT "FOUND" MESSAGE.
016630  004737  016662  2$:     JSR     PC,50$                  ;GO REPORT MESSAGE.

016634  012701  012327          MOV     #EM1406,R1              ;SELECT PROC_2 MESSAGE.
016640  012702  012342          MOV     #EM1407,R2              ;SELECT THE "NOT FOUND" MESSAGE.
016644  120405                  CMPB    R4,R5                   ;CHECK PROC_2 ROM VERSION NUMBER.
016646  001402                  BEQ     4$                      ;GO REPORT PROC_2 CODE NOT FOUND.
016650  012702  012354          MOV     #EM1408,R2              ;SELECT "FOUND" MESSAGE.
016654  004737  016662  4$:     JSR     PC,50$                  ;GO REPORT THE MESSAGE.
016660  000413                  BR      60$                     ;EXIT.

016662                  50$:    PRINTX  #EF1402,R1,R2           ;REPORT THE MESSAGE.
016662  010246                                                          MOV     R2,-(SP)
016664  010146                                                          MOV     R1,-(SP)
```

GLOBAL ERROR REPORTING ROUTINE - ER1401 -

```
        016666  012746  004316                                                  MOV     #EF1402,-(SP)
        016672  012746  000003                                                  MOV     #3,-(SP)
        016676  010600                                                          MOV     SP,R0
        016700  104415                                                          TRAP    C$PNTX
        016702  062706  000010                                                  ADD     #10,SP
  2073  016706  000207                    RTS     PC              ;RETURN.
  2074  016710                    60$:    ENDMSG
        016710                                                          L10006:
        016710  104423                                                          TRAP    C$MSG
```

GLOBAL ERROR REPORTING ROUTINE - ER1601 - SEQ 0053

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER1601 -
                        ;*************************************************************************
                        ;*      This an error reporting subroutine which prints additional error
                        ;*      information if an error is detected in one of the device register
                        ;*      access tests.
                        ;*      This subroutine reports the actual and expected from the device
                        ;*      register(s) which is(are) in faulty.
                        ;*
                        ;* INPUTS:      R1 - Actual data (unused bits set to 0).
                        ;*              R2 - Expected data (unused bits set to 0).
                        ;*              R3 - Offset (in bytes) to the register being tested.
                        ;*              R5 - Line number of register being tested.
                        ;*              RMATBB - Label at base of register message address table.
                        ;*
                        ;* OUTPUTS:     Messages are printed at the operators console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the lable "ER1601" as the message pointer
                        ;*                      parameter in the DRS error report macro call.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE
                        ;*************************************************************************
                                BGNMSG  ER1601
016712                                                                  ER1601::
016712
                                 MOV    RMATBB(R3),R4   ;FETCH ADDRESS OF REGISTER NAME MESSAGE.

                                PRINTB  #EF1604,R4,R5   ;REPORT BASIC MESSAGE (REG NAME AND LINE #).
016716                                                          MOV     R5,-(SP)
016716  010546                                                  MOV     R4,-(SP)
016720  010446                                                  MOV     #EF1604,-(SP)
016722  012746  004503                                         MOV     #3,-(SP)
016726  012746  000003                                         MOV     SP,R0
016732  010600                                                  TRAP    C$PNTB
016734  104414                                                  ADD     #10,SP
016736  062706  000010
                                PRINTX  #EF1602,R2      ;PRINT THE EXPECTED DATA.
016742                                                          MOV     R2,-(SP)
016742  010246                                                  MOV     #EF1602,-(SP)
016744  012746  004377                                         MOV     #2,-(SP)
016750  012746  000002                                         MOV     SP,R0
016754  010600                                                  TRAP    C$PNTX
016756  104415                                                  ADD     #6,SP
016760  062706  000006
                                PRINTX  #EF1603,R1      ;PRINT THE ACTUAL DATA.
016764                                                          MOV     R1,-(SP)
016764  010146                                                  MOV     #EF1603,-(SP)
016766  012746  004441                                         MOV     #2,-(SP)
016772  012746  000002                                         MOV     SP,R0
016776  010600                                                  TRAP    C$PNTX
017000  104415                                                  ADD     #6,SP
017002  062706  000006
                                ENDMSG
017006                                                  L10007:
017006                                                          TRAP    C$MSG
017006  104423
```

GLOBAL ERROR REPORTING ROUTINE - ER1603 -

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER1603 -
                        ;**************************************************************************
                        ;*      This error reporting routine is used to print out a basic error
                        ;*      message, along with a message informing the operator which test is
                        ;*      about to be aborted.
                        ;*
                        ;* INPUTS:      R1 - Contains the address of the message to be printed.
                        ;*              ERRMSG - Contains the address of the message that indicates
                        ;*              the test that is being performed, eg DMA, BREAK etc.
                        ;*
                        ;* OUTPUTS:     Messages are printed at the operators console.
                        ;*              "testname TEST ABORTED"
                        ;*
                        ;* CALLING SEQUENCE:    Include the lable "ER1603" as the message pointer
                        ;*                      parameter in the DRS error report macro call.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;**************************************************************************
017010                          BGNMSG  ER1603
017010                                                                  ER1603::
017010                          SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
017010  004537  003776                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

017014                           PRINTB #EF0503,R1      ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
017014  010146                                                          MOV     R1,-(SP)
017016  012746  004134                                                  MOV     #EF0503,-(SP)
017022  012746  000002                                                  MOV     #2,-(SP)
017026  010600                                                          MOV     SP,R0
017030  104414                                                          TRAP    C$PNTB
017032  062706  000006                                                  ADD     #6,SP

017036  013702  003772           MOV     ERRMSG,R2       ;GET THE "TEST MESSAGE".
017042                           PRINTB #EF1601,R2       ;PRINT "TEST ABORTED" MESSAGE.
017042  010246                                                          MOV     R2,-(SP)
017044  012746  004353                                                  MOV     #EF1601, (SP)
017050  012746  000002                                                  MOV     #2, (SP)
017054  010600                                                          MOV     SP,R0
017056  104414                                                          TRAP    C$PNTB
017060  062706  000006                                                  ADD     #6,SP

017064                          PASS            ;RESTORE THE CONTENTS OF THE GPRS.
017064  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
017066                          ENDMSG
017066                                                                  L10010:
017066  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER3001 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which is intended for use in the
                        ;*      Interrupt BR Level Test.  It reports additional information when an
                        ;*      interrupt has occurred at the wrong BR level.
                        ;*
                        ;* INPUTS:      R1 - Address of message to print first.
                        ;*              R4 - BR level at which the int request occurred.
                        ;*              R5 - Expected or correct BR level for the DUT.
                        ;*
                        ;* OUTPUTS:     A messages is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER3001" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The message is printed as Basic and Extended error information.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*************************************************************************

017070                          BGNMSG  ER3001
017070                                                                  ER3001::
017070                          PRINTB  #EF0503,R1      ;PRINT THE FIRST LINE OF THE MESSAGE.
017070  010146                                                          MOV     R1,-(SP)
017072  012746  004134                                                  MOV     #EF0503,-(SP)
017076  012746  000002                                                  MOV     #2,-(SP)
017102  010600                                                          MOV     SP,R0
017104  104414                                                          TRAP    C$PNTB
017106  062706  000006                                                  ADD     #6,SP
017112                          PRINTX  #EF3001,R5      ;REPORT EXPECTED BR LEVEL.
017112  010546                                                          MOV     R5,-(SP)
017114  012746  004600                                                  MOV     #EF3001,-(SP)
017120  012746  000002                                                  MOV     #2,-(SP)
017124  010600                                                          MOV     SP,R0
017126  104415                                                          TRAP    C$PNTX
017130  062706  000006                                                  ADD     #6,SP
017134                          PRINTX  #EF3002,R4      ;REPORT ACTUAL BR LEVEL.
017134  010446                                                          MOV     R4,(SP)
017136  012746  004647                                                  MOV     #EF3002,-(SP)
017142  012746  000002                                                  MOV     #2,-(SP)
017146  010600                                                          MOV     SP,R0
017150  104415                                                          TRAP    C$PNTX
017152  062706  000006                                                  ADD     #6,SP

017156                          ENDMSG
017156                                                          L10011:
017156  104423                                                          TRAP    C$MSG
```

GLOBAL ERROR REPORTING ROUTINE - ER9004 -

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9004 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which reports error summaries
                        ;*      for lines which have exceeded the specified maximum number of
                        ;*      individual reception errors.
                        ;*
                        ;* INPUTS:      R1 - Address of message to print first.
                        ;*              ERCNTB - Label at base of line error counters table.
                        ;*              ERSMRF - "Report error summary for line" flags.
                        ;*
                        ;* OUTPUTS:     A message is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER9004" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The message is printed as Basic and Extended error information.
                        ;*              The contents of GPR's R2, R3, R4, and R5 are destroyed.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*************************************************************************

017160                          BGNMSG  ER9004
017160                                                                  ER9004::

                        PRINTB  #EF0503,#EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
017160  012746  015315                                                  MOV     #EM9014,-(SP)
017164  012746  004134                                                  MOV     #EF0503,-(SP)
017170  012746  000002                                                  MOV     #2,-(SP)
017174  010600                                                          MOV     SP,R0
017176  104414                                                          TRAP    C$PNTB
017200  062706  000006                                                  ADD     #6,SP
017204  005002                  CLR     R2              ;CLEAR THE LINE COUNTER.
017206  013703  002462          MOV     ERSMRF,R3       ;GET THE ERROR SUMMARY FLAGS.
017212  005004                  CLR     R4              ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
017214  000241          2$:     CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
017216  006003                  ROR     R3              ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
017220  103013                  BCC     4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
017222                          PRINTX  #EF9010,R2,ERCNTB(R4)
017222  016446  002464                                                  MOV     ERCNTB(R4),-(SP)
017226  010246                                                          MOV     R2,-(SP)
017230  012746  005225                                                  MOV     #EF9010,-(SP)
017234  012746  000003                                                  MOV     #3,-(SP)
017240  010600                                                          MOV     SP,R0
017242  104415                                                          TRAP    C$PNTX
017244  062706  000010                                                  ADD     #10,SP
017250  012405          4$:     MOV     (R4)+,R5        ;INCREMENT THE LINE OFFSET BY 2.
017252  005202                  INC     R2              ;INCREMT THE LINE COUNTER.
017254  005703                  TST     R3              ;CHECK THE ERROR SUMMARY FLAGS.
017256  001356                  BNE     2$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.

017260                          ENDMSG
017260                                                          L10012:
017260  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9007 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which is used to report that
                        ;*      something other than a selftest code was found in a selftest code
                        ;*      FIFO slot during the removal of the selftest codes from the FIFO.
                        ;*      This routine is used by the RSTRPT routine.
                        ;*
                        ;* INPUTS:      R1 - Address of error message qualifier st~ing.
                        ;*              R2 - Incorrect code as read from the selftest code FIFO slot.
                        ;*              R3 - Line number associated with the selftest FIFO slot.
                        ;*
                        ;* OUTPUTS:     A message is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER9007" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The message is printed as Basic and Extended error information.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*************************************************************************

017262                          BGNMSG  ER9007
017262                                                                  ER9007::

017262  042703  177760          BIC     #177760,R3      ;REMOVE ALL BUT LINE # BITS FROM LINE # WORD.
017266                          PRINTB  #EF9018,R1,R3   ;REPORT SECONDARY ERROR MESSAGE.
017266  010346                                                          MOV     R3,-(SP)
017270  010146                                                          MOV     R1,-(SP)
017272  012746  005475                                                  MOV     #EF9018,-(SP)
017276  012746  000003                                                  MOV     #3,-(SP)
017302  010600                                                          MOV     SP,R0
017304  104414                                                          TRAP    C$PNTB
017306  062706  000010                                                  ADD     #10,SP
017312                          PRINTX  #EF9017,R1,R2   ;REPORT THE ACTUAL INCORRECT CODE.
017312  010246                                                          MOV     R2,-(SP)
017314  010146                                                          MOV     R1,-(SP)
017316  012746  005421                                                  MOV     #EF9017,-(SP)
017322  012746  000003                                                  MOV     #3,-(SP)
017326  010600                                                          MOV     SP,R0
017330  104415                                                          TRAP    C$PNTX
017332  062706  000010                                                  ADD     #10,SP

017336                          ENDMSG
017336                                                          L10013:
017336  104423                                                          TRAP    C$MSG
```

GLOBAL ERROR REPORTING ROUTINE - ER9008 -

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9008 -
                        ;*****************************************************************************
                        ;*      This is an error reporting subroutine which is used to report that
                        ;*      an unexpected code or character has been found in the DUT receive
                        ;*      character FIFO.
                        ;*
                        ;* INPUTS:      R1 - Address of partial error message string.
                        ;*              R2 - Incorrect code as read from the selftest code FIFO slot.
                        ;*
                        ;* OUTPUTS:     A message is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER9008" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The message is printed as Basic and Extended error information.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*****************************************************************************

017340                          BGNMSG  ER9008
017340                                                                  ER9008::

                        ;*
                        ; Extract the line number from the incorrect code or character which was read
                        ;    from the selftest code FIFO slot.
                        ;-
017340  010203                   MOV    R2,R3
017342  000303                   SWAB   R3
017344  042703  177760           BIC    #177760,R3      ;CALCULATE LINE NUMBER OF CODE.
017350                          PRINTB  #EF9016,R1,R3   ;REPORT TYPE OF INCORRECT CODE FOUND.
017350  010346                                                  MOV     R3,-(SP)
017352  010146                                                  MOV     R1,-(SP)
017354  012746  005324                                         MOV     #EF9016,-(SP)
017360  012746  000003                                         MOV     #3,-(SP)
017364  010600                                                  MOV     SP,R0
017366  104414                                                  TRAP    C$PNTB
017370  062706  000010                                         ADD     #10,SP
017374                          PRINTX  #EF9017,R1,R2   ;REPORT THE ACTUAL INCORRECT CODE.
017374  010246                                                  MOV     R2,-(SP)
017376  010146                                                  MOV     R1,-(SP)
017400  012746  005421                                         MOV     #EF9017,-(SP)
017404  012746  000003                                         MOV     #3,-(SP)
017410  010600                                                  MOV     SP,R0
017412  104415                                                  TRAP    C$PNTX
017414  062706  000010                                         ADD     #10,SP

017420                          ENDMSG
017420                                                  L10014:
017420  104423                                                  TRAP    C$MSG
```

GLOBAL ERROR REPORTING ROUTINE - ER9101 -

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9101 -
                        ;*************************************************************************
                        ;*      This is a general error reporting subroutine which reports a message
                        ;*      which takes a single, 2 digit decimal argument after the end of an
                        ;*      ASCII message.
                        ;*
                        ;* INPUTS:      R1 - Value to be printed after msg as 2 decimal digits.
                        ;*              R2 - Address of message to print first.
                        ;*
                        ;* OUTPUTS:     A messages is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER9101" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The message is printed as Basic error information.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*************************************************************************

017422                          BGNMSG  ER9101
017422                                                          ER9101::

                                PRINTB  #EF9006,R2,R1   ;REPORT THE STRING FOLLOWED BY THE NUMBER.
017422  010146                                                          MOV     R1,-(SP)
017424  010246                                                          MOV     R2,-(SP)
017426  012746  005201                                                  MOV     #EF9006,-(SP)
017432  012746  000003                                                  MOV     #3,-(SP)
017436  010600                                                          MOV     SP,R0
017440  104414                                                          TRAP    C$PNTB
017442  062706  000010                                                  ADD     #10,SP

017446                          ENDMSG
017446                                                          L10015:
017446  104423                                                          TRAP    C$MSG
```

```
 2293                                .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9301 -
 2294                                ;*************************************************************************
 2295                                ;*      This is an error reporting subroutine which prints any BMP codes
 2296                                ;*      that are found in the BMP code queue, together with the the number of
 2297                                ;*      the test that was executing at the time the BMP code was logged.
 2298                                ;*
 2299                                ;* INPUTS:      R1 - The address of the first message to be reported.
 2300                                ;*              R2 - The address of the next empty cell in the queue.
 2301                                ;*
 2302                                ;* OUTPUTS:     The test number followed by the BMP code are printed at the
 2303                                ;*              operator console.
 2304                                ;*
 2305                                ;* CALLING SEQUENCE:    Include the label "ER9301" as the message pointer
 2306                                ;*                      parameter in the Diag Super error report macro call.
 2307                                ;*
 2308                                ;* COMMENTS:    The message is printed as Basic error information.
 2309                                ;*
 2310                                ;* SUBORDINATE ROUTINES USED: None.
 2311                                ;*************************************************************************
 2312
 2313 017450                            BGNMSG  ER9301
      017450                                                                    ER9301::
 2314 017450                            SAVE                    ;SAVE THE GPRS ON THE STACK.
      017450  004537  003776                    JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
 2315
 2316 017454                            PRINTB  #EF0503,R1      ;REPORT UNEXPECTED BMP CODES FOUND.
      017454  010146                                                            MOV     R1,-(SP)
      017456  012746  004134                                                    MOV     #EF0503,-(SP)
      017462  012746  000002                                                    MOV     #2,-(SP)
      017466  010600                                                            MOV     SP,R0
      017470  104414                                                            TRAP    C$PNTB
      017472  062706  000006                                                    ADD     #6,SP
 2317 017476  012703  002526                MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
 2318 017502  012705  015716                MOV     #EM9302,R5      ;GET THE MESSAGE TO BE REPORTED.
 2319 017506  012301                2$:     MOV     (R3)+,R1        ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
 2320 017510  012304                MOV     (R3)+,R4        ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
 2321 017512  004737  017574        JSR     PC,50$          ;GO REPORT THE BMP CODE.
 2322 017516  020302                CMP     R3,R2           ;CHECK IF ALL CODES HAVE BEEN REPORTED.
 2323 017520  103772                BLO     2$              ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
 2324                                ;+
 2325                                ; Check if overflow has occurred.
 2326                                ;  The conditions for overflow are: the pointer contains the address of the
 2327                                ;  last cell in the queue, and a bmp code has already been written into that
 2328                                ;  cell.
 2329                                ;-
 2330 017522  020227  002722                CMP     R2,#BMPCQE-4    ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
 2331 017526  001036                BNE     60$             ;EXIT IF NOT AT THE LAST LOCATION.
 2332 017530  005762  000002        TST     2(R2)           ;CHECK FOR A BMP CODE IN THE LAST CELL
 2333 017534  001433                BEQ     60$             ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
 2334 017536  012301                MOV     (R3)+,R1        ;GET THE TEST NUMBER OFF THE QUEUE.
 2335 017540  011304                MOV     (R3),R4         ;GET THE BMP CODE OFF THE QUEUE.
 2336 017542  012705  015746        MOV     #EM9303,R5      ;SELECT THE MESSAGE TO BE REPORTED.
 2337 017546                        PRINTX  #EF9302         ;REPORT OVERFLOW CONDITION.
      017546  012746  005704                                                    MOV     #EF9302,-(SP)
      017552  012746  000001                                                    MOV     #1,-(SP)
      017556  010600                                                            MOV     SP,R0
      017560  104415                                                            TRAP    C$PNTX
```

GLOBAL ERROR REPORTING ROUTINE - ER9301 - SEQ 0061

```
      017562  062706  000004                                                    ADD     #4,SP
 2338 017566  004737  017574          JSR     PC,50$          ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
 2339 017572  000414                  BR      60$             ;EXIT.
 2340
 2341 017574                  50$:    PRINTX  #EF9301,R5,R1,R4  ;PRINT THE MESSAGE.
      017574  010446                                                    MOV     R4,-(SP)
      017576  010146                                                    MOV     R1,-(SP)
      017600  010546                                                    MOV     R5,-(SP)
      017602  012746  005626                                            MOV     #EF9301,-(SP)
      017606  012746  000004                                            MOV     #4,-(SP)
      017612  010600                                                    MOV     SP,R0
      017614  104415                                                    TRAP    C$PNTX
      017616  062706  000012                                            ADD     #12,SP
 2342 017622  000207                  RTS     PC              ;RETURN.
 2343 017624                  60$:    PASS                    ;RESTORE THE GPR CONTENTS.
      017624  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
 2344
 2345 017626                  ENDMSG
      017626                                                    L10016:
      017626  104423                                                    TRAP    C$MSG
```

GLOBAL SUBROUTINES SECTION

```
2347                            .SBTTL  GLOBAL SUBROUTINES SECTION
2349                            ;**********************************************************************
2350                            ;
2351                            ;               FVTSKL3.P11
2352                            ;
2353                            ;**********************************************************************
2355
2356
2357                            ;++
2358                            ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2359                            ; THAT ARE USED IN MORE THAN ONE TEST.
2360                            ;--
```

```
                                .SBTTL  GLOBAL SUBROUTINE               - ALTFLD -
                                ;++ ******************************************************************************
                                ;*              - Alter Device Register Fields Routine -
                                ;*      This subroutine alters the specified field of the specified device
                                ;*      register for the specified lines.  This routine can be used to set
                                ;*      or clear bits within selected fields of selected registers.
                                ;*      Use examples:   Set RX.BAUD.RATE fields on lines 3 and 6.
                                ;*                      Clear TX.DMA bits on all lines.
                                ;*
                                ;* INPUTS:      R1 - Address of the registers to alter.
                                ;*              R2 - Bit fields set to desired states.
                                ;*              R3 - Bit map of lines for which to alter register
                                ;*              R4 - Mask of bits to alter (1 indicates change bit).
                                ;*              CSRA - Contains the address of the device CSR.
                                ;*              IESTAT - Saved states of the interrupt enable bits.
                                ;*
                                ;* OUTPUTS:     DEVICE REGISTERS - Specified register fields altered.
                                ;*                  CSR IND.ADR.REG field - Destroyed.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,ALTFLD
                                ;*
                                ;* COMMENTS:    This routine reads the specified registers for all lines
                                ;*              with numbers lower than the highes specified line.
                                ;*              This routine does not read the CSR.
                                ;*
                                ;* SUBROUTINES CALLED: None.
                                ;-- ******************************************************************************

017630                          ALTFLD:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017630  004537  003776                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

                                ;+
                                ; Set up to loop for each line:
                                ;    Prepare the word to be ORed into the register contents.
                                ;    Set up the word to write into the IND.ADR.REG field of the CSR.
                                ;-
017634  010400                          MOV     R4,R0           ;CALCULATE THE NEW CONTENTS OF THE
017636  005100                          COM     R0              ;  REGISTER FIELDS WHICH ARE TO BE
017640  040002                          BIC     R0,R2           ;  ALTERRED BY THIS ROUTINE.
017642  013705  002330                  MOV     IESTAT,R5       ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.
                                ;+
                                ; Loop once for each line, altering the specified field in the specified
                                ;    register if the line has been selected for altering.
                                ; Exit the loop if no more lines to alter, or if we have altered the max
                                ;    allowable number of lines (as specified by NUMLNS).
                                ;-
017646  000241                  2$:     CLC                             ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
017650  006003                          ROR     R3              ;GET THE LINE SELECT BIT FOR THIS LINE.
017652  103006                          BCC     4$              ;SKIP SETUP IF LINE IS NOT SELECTED.
017654  010577  162364                  MOV     R5,@CSRA        ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
017660  011100                          MOV     (R1),R0         ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
017662  040400                          BIC     R4,R0           ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
017664  050200                          BIS     R2,R0           ;OR IN THE NEW STATES OF THE FIELDS.
017666  010011                          MOV     R0,(R1)         ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
017670  005205                  4$:     INC     R5              ;SET LINE NUMBER TO THE NEXT LINE.
017672  005703                          TST     R3              ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
017674  001365                          BNE     2$              ;LOOP IF SELECTED LINE(S) IS NOT HANDLED
```

GLOBAL SUBROUTINE - ALTFLD -

```
2418
2419 017676                    60$:    PASS                    ;RESTORE GPRS.
     017676  004736                            JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2420 017700  000207                    RTS     PC              ;RETURN TO CALLING ROUTNE.
```

GLOBAL SUBROUTINE - CALMSL -

```
                                .SBTTL  GLOBAL SUBROUTINE               - CALMSL -
                                ;++ ************************************************************************
                                ;*              - Calibrate Milli Second Loop count subroutine -
                                ;*      This subroutine calibrates the timing loop which is used in the MSLOOP
                                ;*      routine.  This subroutine calculates a value for the MSLCNT variable
                                ;*      which is the number of software loops which takes 1 ms to execute in
                                ;*      the MSLOOP routine.  This routine calibrates the count by using the
                                ;*      Line Time Clock (LTC), so if no LTC is available the default value for
                                ;*      the delay count must be used.
                                ;*
                                ;*
                                ;* INPUTS:      MSLCNT - Default 1 ms delay loop count value, or
                                ;*                       value from previous calibration.
                                ;*              MSTICK - Number of MS per LTC clock tick.
                                ;*              TIMER1 - Timer counter changed by LTC interrupt service rtn.
                                ;*              CLKHRZ - Number of LTC clicks per second (50 or 60).
                                ;*
                                ;* OUTPUTS:     CARRY - Set if LTC is available, and new calibration performed.
                                ;*              MSLCNT - New 1 ms delay loop count value if LTC available, or
                                ;*                       unchanged if no LTC is available.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,CALMSL
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: UNSDIV,OOPS.
                                ;-- ************************************************************************

017702                          CALMSL:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
017702  004537  003776                         JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
017706  005037  020122                 CLR     62$             ;CLEAR THE 2ND TIME FLAG.
                                ;*
                                ; Synchronize with the LTC.
                                ;-
017712  012705  000001          2$:     MOV     #1,R5           ;SET OUTER LOOP COUNTER TO 1 LOOP.
                                                        ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE <**
                                                        ;FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <**
017716  005000                         CLR     R0              ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
017720  012737  000001  002364         MOV     #1,TIMER1       ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
017726  005737  002364          4$:     TST     TIMER1          ;CHECK FOR COUNTER HAVING GONE TO ZERO.
017732  001410                         BEQ     6$              ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
017734  005200                         INC     R0              ;COUNT THIS ITERATION OF THE INNER LOOP.
017736  001373                         BNE     4$              ;LOOP IF COUNTER HAS NOT TURNED OVER.
017740  005305                         DEC     R5              ;DECREMENT THE INNER LOOP COUNTER.
017742  003371                         BGT     4$              ;LOOP IF OUTER LOOP COUNT NOT UP.
                                    ;*
                                    ; If we got no LTC interrupt, indicate that there is no LTC available.
                                    ; LTC must be flakey, or not really an LTC at all.
                                    ;-
017744  005037  002362                 CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC
017750  000241                         CLC                     ;INDICATE FAILURE FOR RETURN.
017752  000461                         BR      60$             ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
                                ;*
                                ; We are now synchronized with the LTC.
                                ; Set up for the calibration loop.
                                ;-
017754  012704  002364          6$:.    MOV     #TIMER1,R4      ;WILL TEST TIMER1 IN THE LOOP BELOW
```

GLOBAL SUBROUTINE - CALMSL -

```
2478 017760  005001                    CLR     R1              ;CLEAR THE OUTER LOOP COUNTER.
2479 017762  005002                    CLR     R2              ;INDICATE TO CHECK ALL BITS OF TIMER1.
2480 017764  005003                    CLR     R3              ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2481 017766  012714  000001            MOV     #1,(R4)         ;LOAD TIMER1 WITH COUNT OF 1.
2482
2483 017772  013705  002376    8$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2484 017776  011400            10$:    MOV     (R4),R0         ;GET THE TIMER1 VALUE.
2485 020000  010037  020124            MOV     R0,64$          ;SAVE WORD (LIKE IN THE REAL LOOP).
2486 020004  040200                    BIC     R2,R0           ;LEAVE ALL THE BITS.
2487 020006  020003                    CMP     R0,R3           ;COMPARE AGAINST ZERO.
2488 020010  000261                    SEC                     ;SET CARRY IN CASE OF SUCCESS.
2489 020012  001406                    BEQ     12$             ;EXIT LOOP IF TIMER1 HAS CLEARED.
2490 020014  005305                    DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2491 020016  001367                    BNE     10$             ;LOOP IF MS NOT UP.
2492 020020  005301                    DEC     R1              ;DECREMENT THE MS TIME COUNT.
2493 020022  001363                    BNE     8$              ;KEEP LOOPING.
2494 020024  004737  020470            JSR     PC,OOPS         ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
2495                            ;+
2496                            ; We have now have loop count information for one clock tick.
2497                            ; We have negative of number of outer loops in R1, each is MSLCNT inner loops.
2498                            ; We have the portion of the last outer loop not executed, in R5.
2499                            ; Now we calculate the total number of inner loops executed.
2500                            ;-
2501 020030  005401    12$:     NEG     R1              ;GET NUMBER OF OUTER LOOPS.
2502 020032  013702  002376            MOV     MSLCNT,R2       ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2503 020036  010203                    MOV     R2,R3           ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2504 020040  160502                    SUB     R5,R2           ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2505 020042  010204                    MOV     R2,R4           ;  AND ADD TO ACCUMULATOR LSWORD.
2506 020044  005005                    CLR     R5              ;CLEAR ACCUMULATOR MSWORD.
2507 020046  005301    14$:     DEC     R1              ;CHECK R1 FOR 0 CONDITION
2508 020050  100403                    BMI     16$             ; SKIP MULTIPLICATION IF ZERO
2509 020052  060304                    ADD     R3,R4           ;MULTIPLY NUMBER OF INNER
2510 020054  005505                    ADC     R5              ; LOOPS PER OUTER LOOP BY
2511 020056  000773                    BR      14$             ;NUMBER OF OUTER LOOPS PERFORMED.
2512                            ;+
2513                            ; Divide the total number of inner loops by the number of MS per LTC tick.
2514                            ;
2515 020060  013701  002374    16$:     MOV     MSTICK,R1       ;# OF MS PER LTC TICK IS DIVISOR.
2516 020064  010403                    MOV     R4,R3           ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2517 020066  010502                    MOV     R5,R2           ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2518 020070  004737  023036            JSR     PC,UNSDIV       ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
2519 020074  103402                    BCS     18$             ;BYPASS OOPS IF WE'RE OK.
2520 020076  004737  020470            JSR     PC,OOPS         ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2521 020102  010137  002376    18$:     MOV     R1,MSLCNT       ;SET NEW VALUE FOR MS LOOP COUNT.
2522 020106  005137  020122            COM     62$             ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2523 020112  001277                    BNE     2$              ;BRANCH IF ONLY ONE ITERATION DONE.
2524 020114  000261                    SEC                     ;SET THE SUCCESS FLAG FOR EXIT.
2525
2526 020116    60$:     PASS                                    ;RESTORE GPRS,
     020116  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
2527 020120  000207                    RTS     PC              ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
2528
2529 020122  000000    62$:     .WORD   0               ;2ND CALIBRATION ITERATION FLAGS.
2530 020124  000000    64$:     .WORD   0               ;DUMMY WORD FOR STORAGE OF THE READ WORD.
```

GLOBAL SUBROUTINE - CKTRAP -

```
                        .SBTTL  GLOBAL SUBROUTINE              - CKTRAP -
                        ;*************************************************************************
                        ;*      Check Trap Routine -
                        ;*      This subroutine is used to check for a bus time-out trap (004 trap)
                        ;*      which is caused by an access to a non-existent memory or I/O location.
                        ;*      If the trap does not occur, this routine returns a success indication.
                        ;*
                        ;* INPUTS:      R0 - Source address for move.
                        ;*              R1 - Destination address for move.
                        ;*              (R0) - Source for the move.
                        ;*
                        ;* OUTPUTS:     (R1) - Written to the contents of (R0).
                        ;*              Carry flag - Set on return if no 004 trap detected.
                        ;*              TP4FLG - Nonzero if trap occurred, cleared otherwise.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CKTRAP
                        ;*
                        ;* COMMENTS:    If this subroutine causes a trap, either the address which
                        ;*              is labeled ADRPTR will be the trap PC address on the stack.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;*************************************************************************

020126                  CKTRAP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020126  004537  003776                      JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
020132  005037  002346              CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAGS.
020136  011011                      MOV     (R0),(R1)       ;PERFORM THE MOVE IN QUESTION.
020140  005737  002346      ADRPTR:: TST    TP4FLG          ;CHECK FOR OCCURENCE OF TRAP.
020144  000261                      SEC                     ;INDICATE SUCCESS.
020146  001401                      BEQ     60$             ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
020150  000241                      CLC                     ;INDICATE FAILURE.
020152                  60$:    PASS                    ;RESTORE GPRS.
020152  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
020154  000207                      RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - CLNRST -
                        ;*****************************************************************************
                        ;*              - Clean Reset of the Device Under Test -
                        ;*      This subroutine is used to reset the DUT to a known state.
                        ;*      The DUT's self-test is skipped,and the fifo is purged of any error
                        ;*      codes, etc.
                        ;*      If the reset does not succesfully complete, then the carry bit is
                        ;*      passed back to the calling routine (clear).
                        ;*
                        ;* INPUTS:      CSRA - Contains the address of the CSR
                        ;*              TXBFCA - Contains address of DUT DMA Buffer Count register.
                        ;*              ERRNBR - Error number for possible error report.
                        ;*              ERRTBL- ERRTYP,ERNBR,and ERRMSG set up correctly.
                        ;*
                        ;* OUTPUTS:     The DUT performs its reset function into a known state.
                        ;*              CARRY - Clear indicates the test is to be aborted.
                        ;*              ERRBLK - value may be destroyed.
                        ;*              IESTAT - TX and RX interrupt flags are cleared.
                        ;*              TX and RX interrupt enable bits in the DUT's CSR are cleared.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CLNRST
                        ;*
                        ;* COMMENTS:    This subroutine can report errors with numbers ERRNBR.
                        ;*              This routine does not destroy the value of ERRNBR.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED. DELAY,MSLGET,PUFIFO,RESETT.
                        ;*****************************************************************************

020156                  CLNRST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020156  004537  003776                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; Reset the DUT.
                        ;  This routine reports errors with numbers from ERRNBR thru ERRNBR+2.
                        ;-
020162  004737  021456          JSR     PC,RESETT       ;RESET THE DUT TO A KNOWN STATE.
020166  103002                  BCC     60$             ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
                        ;+
                        ; Purge the FIFO of error codes, save any BMP codes found.
                        ;
020170  004737  020716          JSR     PC,PUFIFO       ;PURGE THE FIFO.

020174                  60$:                            ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
020174                          PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
020174  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                        ;CARRY BIT:IF CLEAR, THEN ABORT THE TEST.
020176  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - CLR16W -
                        ;++ ******************************************************************************
                        ;*                      - Clear Sixteen Words Routine -
                        ;*      This subroutine clears 16 words starting with the specified word.
                        ;*
                        ;* INPUTS:      R0 - Address of the first word to clear.
                        ;*
                        ;* OUTPUTS:     (R0) to (R0+15) - 16 words of memory are cleared to 0.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CLR16W
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ******************************************************************************

020200                  CLR16W:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020200  004537  003776                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020204  012701  000020              MOV     #16.,R1         ;SET THE LOOP COUNTER TO 16.
020210  005020          2$:         CLR     (R0)+           ;CLEAR A WORD OF MEMORY.
020212  005301                      DEC     R1              ;COUNT THIS LOOP.
020214  001375                    BNE     2$              ;LOOP IF NOT 16 WORD CLEARED.
020216                  60$:    PASS                            ;RESTORE GPRS.
020216  004736                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020220  000207                    RTS     PC
```

GLOBAL SUBROUTINE - CNTERR - SEQ 0070

```
                        .SBTTL  GLOBAL SUBROUTINE               - CNTERR -
                        ;** ************************************************************************
                        ;*                              - Count Error Routine -
                        ;*      This subroutine is used to count a "data" error on the specified
                        ;*      line.  It checks whether error summary reporting is active, or should
                        ;*      be made active on this line, and activates it if necessary.
                        ;*
                        ;* INPUTS:      R5 - Line number of line under consideration.
                        ;*              ERCNTB - Label at base of error counters table.
                        ;*              ERSMRF - Error summary flags (bit set if line in summary mode).
                        ;*              NDERPT - Number of individual data errors to report on a line.
                        ;*
                        ;* OUTPUTS:     CARRY - Set if line is in error summary mode.
                        ;*              ERCNT - Error counter incremented for specified line.
                        ;*              ERSMRF - Bit set if line should be in summary mode.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CNTERR
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ; - ************************************************************************

020222                  CNTERR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020222  004537  003776                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; Count the error on the counter for the specified line.
                        ;-
020226  006305                   ASL     R5                     ;FORM WORD OFFSET FROM LINE NUMBER.
020230  016501  002464           MOV     ERCNTB(R5),R1          ;GET THE PRESENT ERROR COUNT FOR THIS LINE.
020234  005201                   INC     R1                     ;COUNT ERROR.
020236  103402                   BCS     2$                     ;OVERFLOW?  YES, DON'T UPDATE COUNTER IN TABLE.
020240  010165  002464           MOV     R1,ERCNTB(R5)          ;UPDATE ERROR COUNTER TABLE ENTRY.
020244  005737  002230   2$:     TST     NDERPT
020250  001411                   BEQ     60$                    ;SUMMARYS DISABLED?  YES, EXIT WITH CARRY 0.
020252  020137  002230           CMP     R1,NDERPT              ;NO,  CHECK FOR ENOUGH ERRORS FOR SUMMARY USE.
020256  101002                   BHI     4$                     ;ENOUGH ERRORS TO USE SUMMARY?  YES, GO HANDLE.
020260  000241                   CLC                            ;INDICATE NOT TO USE SUMMARY REPORT YET.
020262  000404                   BR      60$                    ;EXIT WITH CARRY 0.
020264  056537  002410  002462 4$:     BIS     BITTBL(R5),ERSMRF       ;SET THE ERROR SUMMARY FLAG FOR LINE.
020272  000261                   SEC                            ;INDICATE TO USE SUMMARY REPORT.
020274                   60$:    PASS                           ;RESTORE GPRS.
020274  004736                                   JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020276  000207                   RTS     PC
```

GLOBAL SUBROUTINE DELAY - SEQ 0071

```
                        .SBTTL  GLOBAL SUBROUTINE                - DELAY -
                        ;*****************************************************************************
                        ;*                                - DELAY SUBROUTINE -
                        ;*      This subroutine is used to delay a variable number of milli-seconds.
                        ;*
                        ;* INPUTS:      R4 - Contains the number of ms to delay.
                        ;*               MSLCNT.
                        ;*
                        ;* OUTPUTS:     None.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,DELAY
                        ;*
                        ;* COMMENTS:    If no hardware clock interrupts are occuring, control-Cs will
                        ;*              not be honored for the duration of the delay.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;*****************************************************************************

020300                  DELAY:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
020300  004537  003776                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020304  010401          MOV     R4,R1   ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
020306  012702  177777  MOV     #-1,R2  ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
020312  005003          CLR     R3      ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
020314  012704  020336  MOV     #62$,R4 ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
020320  004737  020454  JSR     PC,MSLOOP       ;DELAY THE REQUESTED # OF MS.
020324  103002          BCC     60$     ;EXIT ROUTINE IF WE TIMED-OUT.]
020326  004737  020470  JSR     PC,OOPS ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
020332                  60$:    PASS                            ;RESTORE GPRS.
020332  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020334  000207          RTS     PC

020336  177777          62$:    .WORD   -1              ;DUMMY, NON-ZERO WORD.
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - MSLGET -
                        ;*******************************************************************************
                        ;*      - Milli Seconds Loop which returns read word and remaining time -
                        ;*      This subroutine is a general purpose test loop subroutine.  It is used
                        ;*      to verify that a certain action occurs before a time-out period.  The
                        ;*      calling routine passes in which bits should be set and cleared for the
                        ;*      desired condition and the time-out value in milli-seconds.
                        ;*      This routine checks for the desired condition upon entrance into the
                        ;*      routine and then once each milli-second there after.
                        ;*      Upon return, the last word which was read to check for the condition
                        ;*      is returned by this subroutine.
                        ;*
                        ;* INPUTS:      R1 - Time-out value in milli-seconds (up to 64K ms).
                        ;*              R2 - Bit map of bits to test (1 indicates to test the bit).
                        ;*              R3 - Desired states of the indicated fields in R2.
                        ;*              R4 - Address of the word to test.
                        ;*              MSLCNT - Milli second software loop count.
                        ;*
                        ;* OUTPUTS:     R0 - The last word which was read to check for the condition.
                        ;*              R1 - Remaining number of ms in time-out time.
                        ;*              CARRY - Success flag (set if condition is met before time-out).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,MSLGET
                        ;*
                        ;* COMMENTS:    This routine works with or without a hardware clock, but the
                        ;*              calibration is only guarenteed when a line clock is available
                        ;*              on the system.
                        ;*              This routine can be used as a delay routine, by specifying the
                        ;*              desired delay as the time-out and specifying a condition to
                        ;*              look for which will not be met during the delay.
                        ;*              If a time-out value of 0 is specified, this routine checks for
                        ;*              the desired condition before returning.  It indicates success
                        ;*              if the condition is met, failure otherwise.
                        ;*
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;*******************************************************************************

020340                  MSLGET:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
020340  004567  003776                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; Set up mask for removing unused bits in the test word, and clear unused
                        ; bits in the desired state word to allow direct comparison.
                        ;
020344  005102                  COM     R2              ;GET MASK OF UNUSED BITS.
020346  040203                  BIC     R2,R3           ;MASK OUT UNUSED BITS IN DESIRED STATE WORD.
                        ;+
                        ; Handle the test and exit if we have a 0 time-out value.
                        ;
020350  005701                  TST     R1              ;TEST THE TIME-OUT VALUE FOR ZERO.
020352  001011                  BNE     2$              ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
020354  011400                  MOV     (R4),R0         ;GET THE WORD TO TEST BEFORE EXITING.
020356  010037  020452          MOV     R0,62$          ;SAVE VALUE SO WE CAN RETURN IT.
020362  040200                  BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
020364  020003                  CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
020366  000261                  SEC                     ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
020370  001420                  BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
```

GLOBAL SUBROUTINE - MSLGET -

```
2764 020372  000241                    CLC                     ;INDICATE FAILURE (TIME-OUT).
2765 020374  000416                    BR      6$              ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
2766                                ;+
2767                                ; Non-zero time-out value.  Loop, waiting for condition or time-out.
2768                                ;-
2769 020376  013705  002376        2$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2770 020402  011400                4$:     MOV     (R4),R0         ;GET THE WORD TO TEST.
2771 020404  010037  020452                MOV     R0,62$          ;SAVE WORD IN CASE THIS IS THE LAST.
2772 020410  040200                        BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
2773 020412  020003                        CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
2774 020414  000261                        SEC                     ;SET CARRY IN CASE OF SUCCESS.
2775 020416  001405                        BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
2776 020420  005305                        DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2777 020422  001367                        BNE     4$              ;LOOP IF MS NOT UP.
2778 020424  005301                        DEC     R1              ;DECREMENT THE MS TIME COUNT.
2779 020426  001363                        BNE     2$              ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
2780 020430  000241                        CLC                     ;CLEAR CARRY, WE TIMED-OUT.
2781                                ;+
2782                                ; Have either found condition, or timed-out (possibly from 0 time-out value).
2783                                ; Restore the last contents read from the test word.  Exit routine.
2784                                ;-
2785 020432  013700  020452        6$:     MOV     62$,R0          ;PASS OUT THE LAST READ WORD.
2786 020436                        60$:    PASS    R0,R1           ;RESTORE GPRS, EXCEPT THE FOLLOWING:
     020436  010066  000002                        MOV     R0,R0SLOT(SP)       ;PUT R0 IN STACK SLOT.
     020442  010166  000004                        MOV     R1,R1SLOT(SP)       ;PUT R1 IN STACK SLOT.
     020446  004736                                JSR     PC,@(SP)+           ;RETURN TO PREG05 SUBRT.
2787                                                    ;R0 - LAST READ WORD CHECKED FOR CONDITION.
2788                                                    ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
2789 020450  000207                        RTS     PC              ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
2790                                ;+
2791                                ; Local storage.
2792                                ;-
2793 020452  000000                62$:    .WORD   0               ;STORAGE FOR THE LAST READ WORD.
```

GLOBAL SUBROUTINE - MSLOOP -

```
2795                            .SBTTL  GLOBAL SUBROUTINE               - MSLOOP -
2796                            ;*******************************************************************************
2797                            ;*      - Test Loop subroutine -
2798                            ;*      This subroutine is a general purpose test loop subroutine.  It is used
2799                            ;*      to verify that a certain action occurs before a time-out period.  The
2800                            ;*      calling routine passes in which bits should be set and cleared for the
2801                            ;*      desired condition and the time-out value in milli-seconds.
2802                            ;*      This routine checks for the desired condition upon entrance into the
2803                            ;*      routine and then once each milli-second thereafter.
2804                            ;*
2805                            ;* INPUTS:      R1 - Time-out value in milli-seconds (up to 64K ms).
2806                            ;*              R2 - Bit map of bits to test (1 indicates to test the bit).
2807                            ;*              R3 - Desired states of the indicated fields in R2.
2808                            ;*              R4 - Address of the word to test.
2809                            ;*              MSLCNT - Milli second software loop count.
2810                            ;*
2811                            ;* OUTPUTS:     CARRY - Success flag (set if condition is met before time-out).
2812                            ;*
2813                            ;* CALLING SEQUENCE:    JSR     PC,MSLOOP
2814                            ;*
2815                            ;* COMMENTS:    This routine works with or without a hardware clock, but the
2816                            ;*              calibration is only guarenteed when a line clock is available
2817                            ;*              on the system.
2818                            ;*              This routine can be used as a delay routine, by specifying the
2819                            ;*              desired delay as the time-out and specifying a condition to
2820                            ;*              look for which will not be met during the delay.
2821                            ;*              If a time-out value of 0 is specified, this routine checks for
2822                            ;*              the desired condition before returning.  It indicates success
2823                            ;*              if the condition is met, failure otherwise.
2824                            ;*
2825                            ;* SUBORDINATE ROUTINES CALLED: MSLGET.
2826                            ;*******************************************************************************
2827
2828 020454                    MSLOOP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     020454  004537  003776                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2829
2830                            ;+
2831                            ; Calling the MSLGET routine from the MSLOOP routine isolates the caller of
2832                            ; MSLOOP from the returned test word and remaining time-out values.
2833                            ;-
2834 020460  004737  020340            JSR     PC,MSLGET       ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.
2835
2836 020464                    60$:    PASS                            ;RESTORE GPRS,
     020464  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2837 020466  000207                    RTS     PC                        ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
```

```
                                .SBTTL  GLOBAL SUBROUTINE               - OOPS -
                                ;++ ******************************************************************************
                                ;*                          Program abort subroutine -
                                ;*      This subroutine is used to abort the program when a fatal error is
                                ;*      detected in the program or the host system hardware.  An error message
                                ;*      is printed giving some information about the nature of the abort.
                                ;*
                                ;* INPUTS:      R1 - Error code giving reason for abort.
                                ;*
                                ;* OUTPUTS:     An error message is printed.
                                ;*              A list of return PC values for all subroutine calls is printed.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,OOPS
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;-- ******************************************************************************

020470                          OOPS::  SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
020470  004537  003776                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                    ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
020474                              ERRSF   101,EM0101
020474  104454                                                                  TRAP    C$ERSF
020476  000145                                                                  .WORD   101
020500  020534                                                                  .WORD   EM0101
020502  000000                                                                  .WORD   0
                                    ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
020504                              PRINTF  #EM0102
020504  012746  020620                                                          MOV     #EM0102,-(SP)
020510  012746  000001                                                          MOV     #1,-(SP)
020514  010600                                                                  MOV     SP,R0
020516  104417                                                                  TRAP    C$PNTF
020520  062706  000004                                                          ADD     #4,SP
020524                          2$:     BREAK                           ;LOOK FOR OPERATOR CONTROL-C INPUT.
020524  104422                                                                  TRAP    C$BRK
020526  000776                          BR      2$                      ;INFINITE LOOP.
020530                          60$:    PASS                            ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
020530  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020532  000207                          RTS     PC                      ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.

020534    110     117     123   EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
020537    124     040     103
020542    117     115     120
020545    125     124     105
020550    122     040     110
020553    101     122     104
020556    127     101     122
020561    105     040     117
020564    122     040     123
020567    117     106     124
020572    127     101     122
020575    105     040     102
020600    125     107     040
020603    105     116     103
020606    117     125     116
020611    124     105     122
```

GLOBAL SUBROUTINE - OOPS -

```
        020614      105     104     056
        020617      000
   2869 020620      045     116     045  EM0102:: .ASCIZ /%N%APROGRAM HUNG, WAITING FOR A CONTROL-C. <**************%N%N/
        020623      101     120     122
        020626      117     107     122
        020631      101     115     040
        020634      110     125     116
        020637      107     054     040
        020642      127     101     111
        020645      124     111     116
        020650      107     040     106
        020653      117     122     040
        020656      101     040     103
        020661      117     116     124
        020664      122     117     114
        020667      055     103     056
        020672      040     074     052
        020675      052     052     052
        020700      052     052     052
        020703      052     052     052
        020706      052     052     052
        020711      045     116     045
        020714      116     000
   2870                                 .EVEN
```

GLOBAL SUBROUTINE - PUFIFO -

```
                        .SBTTL  GLOBAL SUBROUTINE               - PUFIFO -
                        ;*************************************************************************
                        ;*      - PURGE THE FIFO
                        ;*      This routine tries to remove all the characters from the FIFO.
                        ;*      Any BMP codes that are found are saved on the BMP code queue.
                        ;*
                        ;* INPUTS:      RBUFA- Contains the  address of the Receiver.
                        ;*
                        ;*
                        ;* OUTPUTS:     Carry bit - Indicates the state of the fifo, set:= purged.
                        ;*              BMPCQ - The contents of the BMP code queue may be updated.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,PUFIFO
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
                        ;*************************************************************************

020716                  PUFIFO::SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
020716  004537  003776                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020722  012701  001000          MOV     #512.,R1                ;SET MAXIMUM TRY COUNT OF 512.
020726  013704  002246          MOV     RBUFA,R4                ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.

020732  011402          2$:     MOV     (R4),R2                 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
020734  100016                  BPL     6$                      ;EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.
                        ;*
                        ; Check if the read character is actually a BMP code.
                        ;  If it is, then save it on the BMP code queue to be reported later.
                        ;-
020736  012700  070000          MOV     #70000,R0               ;GENERATE A BIT MAP OF CHAR ERROR BITS
020742  040200                  BIC     R2,R0                   ; WHICH ARE NOT SET FOR CHAR.
020744  001006                  BNE     4$                      ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
                          ;*
                          ; Check if the read data is modem status , BMP or Selftest?.
                          ;-
020746  012700  000300          MOV     #300,R0                 ; CHECK IF BMP OR SELFTEST?.
020752  040200                  BIC     R2,R0                   ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
020754  001002                  BNE     4$                      ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
020756  004737  022262          JSR     PC,SAVBMP               ;SAVE BMP CODE ON THE QUEUE.

020762  005301          4$:     DEC     R1                      ;DECREMENT THE TRY COUNT.
020764  001362                  BNE     2$                      ;LOOP TO TRY AGAIN.
020766  000241                  CLC                             ;CLEAR CARRY,TO INDICATE FIFO NOT PURGED.
020770  000401                  BR      60$                     ;EXIT WITH CARRY CLEAR.
020772  000261          6$:     SEC                             ;SET CARRY, TO INDICATE FIFO PURGED.

020774                  60$:    PASS                            ;RESTORE GPRS,
020774  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;CARRY BIT, SET INDICATES FIFO PURGED.
020776  000207                  RTS     PC
```

GLOBAL SUBROUTINE - RDPDR -

```
                        .SBTTL  GLOBAL SUBROUTINE                - RDPDR -
                        ;*****************************************************************************
                        ;*      - Read and verify Data Pattern from Device Registers Routine -
                        ;*      This routine reads and verifies the rotated data pattern which has
                        ;*      been written by the WDPDR subroutine.
                        ;*      Each active line's register's contents is read and compared with the
                        ;*      written data.
                        ;*      After the unused and Read Only (RO) bits are masked out, any errors are
                        ;*      reported from this routine.
                        ;*      This routine will take into account the type of write operation which
                        ;*      was performed by the WDPDR subroutine.
                        ;*
                        ;* INPUTS:      R2 - Used to pass in the data pattern to be rotated & verified.
                        ;*              R3 - Byte indicator (- => lo byte, + => hi byte, 0 => both).
                        ;*              R4 - Operation type indicator (- => BIC, + => BIS, 0 => MOV).
                        ;*              ACTLNS - Bit map of active lines on the device under test.
                        ;*              CSRA - Contains the CSR address of the Device under test.
                        ;*              DRADRT - Base address of device register address table.
                        ;*              ERCNTB - Label at base of error counters table for lines.
                        ;*              ERRMSG - Set up with the proper error message for this test.
                        ;*              ERRNBR - Set up with the proper error number.
                        ;*              LPR0 - Equated to LPR reg offset from device CSR address.
                        ;*              NUMLNS - Number of lines on the device under test.
                        ;*              NDERPT - Number of individual data errors to report on a line.
                        ;*              TXBFC0 - Equated to TBUFFCT reg offset from device CSR address.
                        ;*              UNBTTB - Base address of the unused bit table.
                        ;*
                        ;* OUTPUTS:     Error messages may be printed at the operator's console.
                        ;*              ERCNT - Error counters table is updated for line under test.
                        ;*              ERRBLK - Contents destroyed.
                        ;*              ERSMRF - Error summary flags bit set if line in summary mode.
                        ;*              UUT CSR - All bits cleared, except IND.ADR.REG field destroyed.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RDPDR
                        ;*
                        ;* COMMENTS:    For byte accesses, only the specified byte is verified.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ER1601,ROLDAP.
                        ;-- *****************************************************************************

021000                  RDPDR:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
021000  004537  003776                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021004  012737  016712  003774          MOV     #ER1601,ERRBLK  ;SET UP THE ADDRESS OF THE ERROR REPORT RTN.
                        ;*
                        ; Determine whether register data should be inverted from data pattern.
                        ;-
021012  005704          TST     R4              ;CHECK THE OPERAND TYPE INDICATOR.
021014  100001          BPL     2$              ;BIC WRITE PERFORMED?  NO, USE STANDARD DATA
021016  005102          COM     R2              ;YES, INVERT THE DATA PATTERN.
                        ;*
                        ; Set up outer loop.
                        ;
021020  005005          2$:     CLR     R5              ;CLEAR LINE COUNTER TO SELECT LINE 0
                        ;*
                        ; The outer loop follows.  Each pass through this loop reads and compares data
                        ; from all of the device registers for a particular line if the line is active.
                        ;
```

GLOBAL SUBROUTINE - RDPDR - SEQ 0079

```
 2978 021022  010237  021216          4$:     MOV     R2,70$          ;SAVE THE OUTER LOOP DATA PATTERN.
 2979 021026  010577  161212                  MOV     R5,@CSRA        ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
 2980 021032  010500                          MOV     R5,R0
 2981 021034  006300                          ASL     R0
 2982 021036  036037  002410  002236          BIT     BITTBL(R0),ACTLNS
 2983 021044  001452                         BEQ     16$             ;IS THE LINE ACTIVE?  NO, SKIP THE LINE.
 2984 021046  012703  000004                  MOV     #LPR0,R3        ;YES, INITIALIZE REGISTER OFFSET FOR LPR.
 2985                                 ;+
 2986                                 ;       The inner loop follows.  Each pass through this loop reads and compares
 2987                                 ;       data from a device register.
 2988                                 ;-
 2989 021052  010204                  6$:     MOV     R2,R4           ;SAVE THE INNER LOOP DATA PATTERN.
 2990 021054  046302  002264                  BIC     UNBTTB(R3),R2   ;REMOVE UNUSED BITS FROM EXPECTED DATA.
 2991 021060  016300  002244                  MOV     DRADRT(R3),R0
 2992 021064  005766  000010                  TST     R3SLOT(SP)      ;CHECK THE ACCESS TYPE INDICATOR.
 2993 021070  001002                         BNE     8$              ;BYTE ACCESS?  YES, GO PERFORM BYTE READ.
 2994 021072  011001                          MOV     (R0),R1         ;NO, PERFORM WORD READ OF DEVICE REGISTER.
 2995 021074  000416                         BR      12$
 2996 021076  100410                  8$:     BMI     10$             ;LOW BYTE ACCESS?  YES, GO DO LOW BYTE READ.
 2997 021100  005200                          INC     R0              ;HIGH BYTE ACCESS.  FORM HIGH BYTE ADDRESS.
 2998 021102  111001                          MOVB    (R0),R1         ;READ THE HI BYTE OF THE DUT REGISTER.
 2999 021104  000301                          SWAB    R1              ;PUT HI BYTE BACK INTO THE HI BYTE.
 3000 021106  042701  000377                  BIC     #377,R1         ;REMOVE THE UNUSED BYTE IN ACTUAL DATA.
 3001 021112  042702  000377                  BIC     #377,R2         ;REMOVE THE UNUSED BYTE IN EXPECTED DATA.
 3002 021116  000405                         BR      12$
 3003 021120  111001                  10$:    MOVB    (R0),R1         ;READ THE LOW BYTE OF THE DUT REGISTER.
 3004 021122  042701  177400                  BIC     #177400,R1      ;REMOVE THE UNUSED BYTE.
 3005 021126  042702  177400                  BIC     #177400,R2      ;FORM EXPECTED LOW BYTE FOR COMAPARISON.
 3006
 3007 021132  046301  002264          12$:    BIC     UNBTTB(R3),R1   ;REMOVE UNUSED BITS FROM ACTUAL DATA.
 3008 021136  020102                          CMP     R1,R2           ;COMPARE ACTUAL AND EXPECTED DATA.
 3009 021140  001404                         BEQ     14$             ;ACTUAL = EXPECTED?  YES, SKIP ERROR.
 3010 021142  004737  020222                  JSR     PC,CNTERR       ;NO, COUNT THE ERROR, CHECK FOR ERROR SUMMARY.
 3011 021146  103401                         BCS     14$             ;USE ERROR SUMMARY?  YES, SKIP ERROR.
 3012                                     ;No, report "BAD BIT(S) IN DEVICE xxxxx REGISTER FOR LINE nn (D)."
 3013 021150                              ERROR
      021150  104460                                                                 TRAP    C$ERROR
 3014 021152  010402                  14$:    MOV     R4,R2           ;RESTORE THE INNER LOOP DATA PATTERN.
 3015 021154  004737  021570                  JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
 3016 021160  062703  000002                  ADD     #2,R3           ;SET REGISTER OFFSET TO THE NEXT REGISTER.
 3017 021164  020327  000016                  CMP     R3,#TXBFCO      ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
 3018 021170  003730                          BLE     6$              ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
 3019                                 ;+
 3020                                 ; Back into the outer loop.  Now set up for next line.  Loop if not done.
 3021                                 ;-
 3022 021172  013702  021216          16$:    MOV     70$,R2          ;SET UP TO ROTATE THE DATA PATTERN.
 3023 021176  004737  021570                  JSR     PC,ROLDAP       ;ROTATE THE DATA PATTERN.
 3024 021202  005205                          INC     R5              ;COUNT THIS LINE
 3025 021204  020527  000010                  CMP     R5,#NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
 3026 021210  002704                          BLT     4$              ;LOOP IF SOME LINES NOT DONE.
 3027
 3028 021212                          60$:    PASS                    ;RESTORE GPRS.
      021212  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
 3029 021214  000207                          RTS     PC
 3030
 3031 021216  000000                  70$:    .WORD   0               ;STORAGE FOR DATA PATTERN OUTSIDE INNER LOOP.
```

GLOBAL SUBROUTINE - REGTST -

```
                        .SBTTL  GLOBAL SUBROUTINE               - REGTST -
                        ;** ************************************************************************
                        ;*                              - Registers Test Subroutine -
                        ;*      Subroutine to test the Device Under Test (DUT) registers.  The used
                        ;*      bits of the registers are either all cleared or all set and then the
                        ;*      data pattern is written and verified using either word or byte
                        ;*      accesses in read/write or read/modify/write mode.
                        ;*
                        ;* INPUTS:      R3 - Byte indicator (- => low, + => high, 0 => both bytes).
                        ;*              R4 - Access mode (-1 => set then BIC, 1 => clear then BIS,
                        ;*                      (-2 => set them MOV, +2 clear then MOV).
                        ;*              ERRNBR - Set up with initial error number.
                        ;*
                        ;* OUTPUTS:     GPRS0 - GPR save area 0 is destroyed.
                        ;*              Device Under Test registers are written.
                        ;*              Error messages may be printed at the operators console.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,REGTST
                        ;*
                        ;* COMMENTS:    This routine loop 16 times writing the same data pattern
                        ;*              rotated left once each iteration.
                        ;*              This routine can report errors INITIAL ERRNBR thru INITIAL+2.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: RDPDR,ROLDAP,SWAP0,WDPDR
                        ; - ************************************************************************

021220                  REGTST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021220  004537  003776                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; Set up the GPRs for the writting of the data pattern.
                        ;-
021224  012705  000020                  MOV     #16.,R5         ;SET UP LOOP COUNTER TO COUNT 16 ITERATIONS.
021230  012702  167410                  MOV     #167410,R2      ;INITIALIZE THE DATA PATTERN.
021234  032704  000001                  BIT     #BIT0,R4        ;TEST FOR R/W ACCESS.
021240  001001                        BNE     2$              ;R/M/W ACCESS?  YES, R4 IS ALL SET UP.
021242  005004                          CLR     R4              ;NO, INDICATE R/W ACCESS.
021244                          2$:
                        ;+
                        ; Set up the GPRs for the clearing or setting of all the used bits.
                        ;-
021244  010400                  MOV     R4,R0           ;PASS OPERATION TYPE INDICATOR AROUND SWAP0.
021246  004737  022406                JSR     PC,SWAP0        ;GET ALTERNATE GPR SET IN R1 THRU R5.
021252  013701  003770                  MOV     ERRNBR,R1       ;SAVE THE INITIAL ERROR NUMBER.
021256  010004                          MOV     R0,R4
021260  005404                          NEG     R4              ;SET UP OP TYPE FOR CLEARING OR SETTING.
021262  005002                          CLR     R2              ;SET UP CLEAR WRITE PATTERN.
021264  026627  000012  000002          CMP     R4SLOT(SP),#2   ;TEST FOR CLEAR THEN MOV TEST SEQUENCE.
021272  001401                        BEQ     4$              ;CLEAR THEN MOV?  YES, LEAVE WRITE PAT CLEAR.
021274  005102                          COM     R2              ;NO, SET ALL BITS OF WRITE PATTERN.
021276  005003                  4$:     CLR     R3              ;INDICATE THAT WORD ACCESSES SHOULD BE USED.
021300  005000                          CLR     R0              ;SET ALTERNATE BYTE EXPECTED DATA PAT TO CLEAR.
021302  026627  000012  177776          CMP     R4SLOT(SP),#-2  ;TEST FOR SET THEN MOV TEST SEQUENCE.
021310  001001                        BNE     6$              ;SET THEN MOV?  YES, LEAVE ALT BYTE PAT CLEAR.
021312  005100                          COM     R0              ;NO, SET ALT BYTE EXPECTED DATA PAT TO ALL 1'S.
021314  004737  022406          6$:     JSR     PC,SWAP0        ;RESTORE SWAPPED GPR VALUES TO R1 THRU R5.
                        ;+
                        ; Start of data pattern loop.
```

```
                                        ;-
                                8$:
                                    ;+
                                    ; Set or clear all the used bits of the device registers for all lines.
                                    ; Verify that all the bits were set or cleared correctly.
                                    ;-
021320  004737  022406                  JSR     PC,SWAPO        ;GET ALTERNATE GPRS FOR SETTING INTIAL STATES.
021324  004737  023322                  JSR     PC,WDPDR        ;GO CLEAR ALL USED REGISTER BITS, ALL LINES.
021330  010137  003770                   MOV    R1,ERRNBR       ;SET UP ERROR NUMBER TO INITIAL ERRNBR.
021334  004737  021000                  JSR     PC,RDPDR        ;VERIFY ALL USED REGISTER BITS, ALL LINES.
021340  004737  022406                  JSR     PC,SWAPO        ;RESTORE MAIN GPRS CONTENTS.
                                    ;+
                                    ; Write data patterns, all lower byte used bits, all registers, all lines.
                                    ; Verify that the data pattern was written correctly.
                                    ;-
021344  004737  023322                  JSR     PC,WDPDR        ;WRITE DATA PATTERN TO DEVICE REGISTERS.
021350  005237  003770                   INC    ERRNBR          ;SET ERROR NUMBER TO INITIAL+1.
021354  004737  021000                  JSR     PC,RDPDR        ;VERIFY DATA PATTERN IN ALTERRED BYTE(S).
021360  005703                           TST    R3              ;CHECK THE BYTE INDICATOR.
021362  001411                          BEQ     10$             ;WORD ACCESS?  YES, SKIP SECOND BYTE CHECK.
                                    ;+
                                    ; Check that the alternate (unmodified) byte is clear or set as expected.
                                    ;-
021364  010201                           MOV    R2,R1           ;SAVE THE DATA PATTERN.
021366  010002                           MOV    R0,R2           ;GET THE ALTERNATE BYTE EXPECTED DATA.
021370  005403                           NEG    R3              ;INDICATE THAT OTHER BYTE IS TO BE CHECKED.
021372  005237  003770                   INC    ERRNBR          ;SET ERROR NUMBER TO INITIAL+2.
021376  004737  021000                  JSR     PC,RDPDR        ;VERIFY DATA PATS IN OTHER BYTES OF REGISTERS.
021402  005403                           NEG    R3              ;RESTORE BYTE INDICATOR.
021404  010102                           MOV    R1,R2           ;RESTORE DATA PATTERN.
                                    ;+
                                    ; Pepare the next data pattern and loop if not done.
                                    ;-
021406  004737  021570          10$:    JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
021412  005305                           DEC    R5              ;COUNT THIS ITERATION OF THE LOOP.
021414  003341                          BGT     8$              ;ALL PATTERNS DONE?  NO, LOOP.
                                                                ;YES, RESTORE ERROR NUMBER AND EXIT.
021416  013737  002450  003770  60$:    MOV     GPRS0B,ERRNBR   ;GET THE ERROR NUMBR FROM GPR SWAP STORAGE.
021424                                  PASS                    ;RESTORE GPRS.
021424  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
021426  000207                          RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - REPSMR -
                                ;** ************************************************************************
                                ;*                      - Report Error Summary Routine -
                                ;*      This subroutine reports an error summary for those lines which have
                                ;*      exceeded the number of individual errors to report for a single line
                                ;*      in a single test.  This parameter can be specified by the operator if
                                ;*      he/she answers the Software Parameter Questions.
                                ;*
                                ;* INPUTS:      ERCNTB - Label at base of line error counters table.
                                ;*              ERRMSG - Address of primary error message.
                                ;*              ERRNBR - Error number of errors in this routine.
                                ;*              ERSMRF - "Report error summary for line" flags.
                                ;*
                                ;* OUTPUTS:     ERRBLK - Address of error reporting routine (Destroyed).
                                ;*              Summary messages may be printed at the operator console.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,REPSMR
                                ;*
                                ;* COMMENTS:    If no lines have exceeded the maximum number of individual
                                ;*              errors to report, no messages are printed by this routine.
                                ;*              Error summaries in this routine are reported as errors.
                                ;*              The contents of ERRBLK are destroyed.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED:
                                ; - ************************************************************************

021430                          REPSMR:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
021430  004537  003776                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021434  005737  002462                  TST     ERSMRF          ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
021440  001404                          BEQ     60$             ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
                                ;*
                                ; We have some error summaries to report.
                                ;-
021442  012737  017160  003774          MOV     #ER9004,ERRBLK  ;SELECT ERROR REPORTING ROUTINE.
                                ;*
                                ; Report
                                ;     "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
                                ;
021450                              ERROR
021450  104460                                                                  TRAP    C$ERROR

021452                          60$:    PASS                    ;RESTORE GPRS.
021452  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
021454  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - RESETT -
                        ;*****************************************************************************
                        ;*              - Reset Device Under Test -
                        ;*      This subroutine is used to reset the DUT to a known state.
                        ;*      If reset does not succesfully complete, ie. time-out occurs, then
                        ;*      an abort test error message is reported.
                        ;*
                        ;* INPUTS:      CSRA - Contains the address of the CSR
                        ;*              TXBFCA - Contains address of DUT DMA Buffer Count register.
                        ;*              ERRTBL- ERRTYP,ERNBR,and ERRMSG set up correctly.
                        ;*
                        ;* OUTPUTS:     The DUT performs its reset function into a known state.
                        ;*              CARRY - Clear indicates the test is to be aborted.
                        ;*              ERRBLK - value may be destroyed.
                        ;*              IESTAT - TX and RX interrupt flags are cleared.
                        ;*              TX and RX interrupt enable bits in the DUT's CSR are cleared.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RESETT
                        ;*
                        ;* COMMENTS:    This subroutine can report errors with numbers initial ERRNBR
                        ;*              This routine does not destroy the value of ERRNBR.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET.
                        ;*****************************************************************************
                        RESETT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
021456  004537  003776                JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
021462  012702  000040        MOV     #BIT05,R2       ;SET BIT MASK OF MASTER RESET BIT.
                        ;*
                        ; Test the state of the master reset bit in the CSR.
                        ;  If MR is set then wait for self-test to complete.
                        ;   If time-out occurs, report the error and pass-out abort test indicator.
                        ;
021466  013704  002244          MOV     CSRA,R4         ;GET THE ADDRESS OF THE DUT'S CSR.
021472  030214          BIT     R2,(R4)         ;CHECK STATE OF MASTER RESET BIT.
021474  001406        BEQ     2$              ;DON'T DELAY IF MR IS ALREADY CLEAR.
021476  005003          CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
021500  012701  004704          MOV     #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
021504  004737  020340        JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
021510  103012          BCC     4$              ;GO REPORT ERROR IF TIMEOUT OCCURRED.
                        ;*
                        ; Set Master Reset bit in CSR.  Clear TX and RX enable bits, etc.
                        ;  Skip the selftest.
                        ;  Time-out of 2.5 secs, just in case the self-test executes.
                        ;-
021512  010277  160526  2$:      MOV     R2,@CSRA        ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
021516  004737  022330        JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
                          ;*
                          ;   Set Self-test time-out of 2.5 seconds, and wait for M.R to clear.
                          ;   If Time out occurs, then report the fatal error and pass out the abort
                          ;   test indicator.
                          ;-
021522  005003          CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
021524  012701  004704          MOV     #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
021530  004737  020340        JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
021534  103410          BCS     6$              ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
```

GLOBAL SUBROUTINE - RESETT -

```
3228                                    ;+
3229                                    ; Set up error message to report "fatal error found during reset,test aborted".
3230                                    ; Indicate test is to be aborted by clearing the carry bit.
3231                                    ;-
3232 021536  012701  012362            4$:     MOV     @EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
3233 021542  012737  017010  003774            MOV     @ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
3234                                            ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
3235                                            ; "TEST ABORTED"
3236 021550                                    ERROR                   ;               >>>>> ERROR <<<<<
     021550  104460                                                                            TRAP    C$ERROR
3237 021552  000241                            CLC                     ;INDICATE TEST IS TO BE ABORTED.
3238 021554  000403                            BR      60$             ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
3239                                    ;+
3240                                    ; Clear TX and RX Interrupt enable status flags in IESTAT.
3241                                    ; Exit with continue test indicator set (ie,carry set).
3242                                    ;-
3243 021556  005037  002330            6$:      CLR    IESTAT          ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
3244 021562  000261                             SEC                    ;INDICATE SUCCESS, CONTINUE TEST.
3245
3246 021564                            60$:    PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
     021564  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
3247                                            ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
3248 021566  000207                            RTS     PC
3249
```

GLOBAL SUBROUTINE - ROLDAP -

```
                                .SBTTL  GLOBAL SUBROUTINE               - ROLDAP -
                                ;*******************************************************************************
                                ;*      - ROtate Left DAta Pattern
                                ;*      This routine rotates the passed input data pattern left,without going
                                ;*      through the carry.The carry is initially set or cleared depending
                                ;*      upon the state of the MSB of the data pattern,before a ROL instruction
                                ;*      is executed.
                                ;*
                                ;* INPUTS:      R2 - Contains the data pattern to be rotated
                                ;*
                                ;* OUTPUTS:     R2 - Contains the rotated data pattern
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,ROLDAP
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE
                                ;*******************************************************************************

021570                          ROLDAP::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
021570  004537  003776                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021574  010202                  MOV     R2,R2           ;SET PROCESSOR STATUS CODES
021576  005702                  TST     R2              ;CHECK MSB
021600  100402                  BMI     2$              ;BRANCH IF SET
021602  000241                  CLC                     ;CLEAR CARRY BIT IF MSB CLEAR
021604  000401                  BR      4$              ;
021606  000261          2$:     SEC                     ;SET CARRY IF MSB SET
021610  006102          4$:     ROL     R2              ;ROTATE DATA PATTERN LEFT
021612                  60$:    PASS    R2              ;RESTORE GPRS,EXCEPT
021612  010266  000006                          MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
021616  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                 ;R2 - CONTAINS THE ROTATED DATA PATTERN
021620  000207                  RTS     PC
```

GLOBAL SUBROUTINE - RSTRPT -

```
                                .SBTTL  GLOBAL SUBROUTINE               - RSTRPT -
                                ;++ ******************************************************************************
                                ;*                      Report any Reset Errors Routine -
                                ;*      This routine determines if any error codes are among the diagnostic
                                ;*      codes reported placed in the DUT received character FIFO by the
                                ;*      Self-test.  If any non BMP error codes are found, or if other errors
                                ;*      are encountered, appropriate errors are reported.  Any BMP codes that
                                ;*      are found, are placed on the BMP code queue to be reported later.
                                ;*      This routine also purges the DUT FIFO looking for any characters
                                ;*      or modem status codes.  If any are found, errors are reported.
                                ;*
                                ;* INPUTS:      ERRMSG - Address of the primary error message.
                                ;*              ERRNBR - Error number of first error reported by this rouitne.
                                ;*              NUMLNS - Equated to the number of line on the DUT.
                                ;*              RBUFA - Contains address of the DUT receiver FIFO.
                                ;*
                                ;* OUTPUTS:     CARRY - Success flag (set if FIFO cleared successfully).
                                ;*              ERRBLK - Address of the error report routine (Destroyed).
                                ;*              Error messages can be printed at the operators console.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,RSTRPT
                                ;*
                                ;* COMMENTS:    This subroutine can report errors with numbers Initial ERRNBR
                                ;*              thru Initial ERRNBR+4.
                                ;*              This routine does not destroy the value of ERRNBR.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: ER0503,ER9007,ER9008,SAVBMP.
                                ;-- ******************************************************************************

021622                          RSTRPT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021622  004537  003776                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;+
                                ; Read correct number (number of line on DUT) of chars from the FIFO.
                                ; Verify that each char is a selftest success code.
                                ;-
021626  005003                          CLR     R3                      ;CLEAR THE CODE COUNTER.
021630  013705  003770                  MOV     ERRNBR,R5               ;SAVE ERRNBR FOR RESTORATION LATER.
021634  017702  160406          2$:     MOV     @RBUFA,R2               ;READ A CHAR FROM THE DUT FIFO.
021640  100412                        BMI     4$                      ;SKIP ERROR IF DATA.VALID SET FOR CHAR.
                                    ;+
                                    ; We expect a selftest code, but this FIFO slot is empty.
                                    ;-
021642  010537  003770                  MOV     R5,ERRNBR               ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
021646  012701  015522                  MOV     #EM9018,R1              ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
021652  012737  017262  003774          MOV     #ER9007,ERRBLK          ;SELECT PROPER ERROR REPORT ROUTINE.
                                    ;+
                                    ; Report error with number Initial ERRNRB.
                                    ;    "NO SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE nn AFTER RESET."
                                    ;-
021660                              ERROR                               ;               >>>>> ERROR <<<<<.
021660  104460                                                                                    TRAP    C$ERROR
                                    ;+
                                    ; Inidicate "success' (because FIFO is purged), and exit this routine.
                                    ;-
021662  000261                          SEC                             ;SET SUCCESS FLAG.
021664  000545                        BR      60$                     ;EXIT ROUTINE.
                                    ;+
```

GLOBAL SUBROUTINE - RSTRPT - SEQ 0087

```
3337                                        ; Determine if this is not a selftest code.
3338                                        ;-
3339 021666  012700  070001          4$:     MOV     #70001,R0       ;GENERATE BIT MAP OF ANY CLEAR ERROR BITS OR
3340 021672  040200                          BIC     R2,R0           ;  BIT 0 WHICH ARE CLEAR.
3341 021674  001033                         BNE     8$              ;GO TO REPORT ERROR IF THIS IS NOT A TEST CODE.
3342                                        ;+
3343                                        ; We have a test code (either BMP or selftest code).
3344                                        ; Determine what type of code we have.
3345                                        ;
3346 021676  032702  000200                  BIT     #BIT7,R2        ;TEST ROM VERSION CODE INDICATOR BIT.
3347 021702  001443                         BEQ     10$             ;SKIP ERRORS IF SELFTEST ROM VERSION CODE.
3348 021704  120227  000203                  CMPB    R2,#203         ;CHECK IF SKIP SELF TEST CODE.
3349 021710  001440                         BEQ     10$             ;SKIP ERROR REPORT IF SKIP SELF TEST CODE FOUND
3350 021712  120227  000201                  CMPB    R2,#201         ;CHECK IF NULL CODE PRESENT.
3351 021716  001435                         BEQ     10$             ;SKIP ERROR REPORT IF SELF TEST NULL CODE.
3352 021720  012700  000300                  MOV     #300,R0         ;TEST CODE TYPE BITS FOR BOTH CODE
3353 021724  040200                          BIC     R2,R0           ;  TYPE BITS SET (INDICATING BMP CODE).
3354 021726  001003                         BNE     6$              ;IF IT IS NOT A BMP CODE GO REPORT ERROR.
3355 021730  004737  022262                  JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
3356 021734  000426                         BR      10$             ;GO GET THE NEXT CHARACTER FROM THE FIFO.
3357                                        ;+
3358                                        ; We have a selftest error code.
3359                                        ;-
3360 021736  010537  003770          6$:     MOV     R5,ERRNBR       ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
3361 021742  005237  003770                  INC     ERRNBR          ;CALCULATE INITIAL ERROR NUMBER PLUS 1.
3362 021746  012701  015547                  MOV     #EM9020,R1      ;PASS ERROR MESSAGE INFO TO ER9008 ROUTINE.
3363 021752  012737  017340  003774          MOV     #ER9008,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3364                                        ;+
3365                                        ; Report error with number Initial ERRNRB + 1.
3366                                        ;    "UNEXPECTED SELFTEST ERROR CODE FOR LINE nn IN FIFO AFTER RESET:"
3367                                        ;-
3368 021760                                  ERROR                   ;               >>>>> ERROR <<<<<.
     021760  104460                                                                       TRAP    C$ERROR
3369 021762  000413                         BR      10$             ;GO TO END OF LOOP.
3370                                        ;+
3371                                        ; We have a non-selftest code (either BMP code or data char).
3372                                        ;-
3373 021764  010537  003770          8$:     MOV     R5,ERRNBR       ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
3374 021770  062737  000002  003770          ADD     #2,ERRNBR       ;CALCULATE INITIAL ERROR NUMBER PLUS 2.
3375 021776  012701  015532                  MOV     #EM9019,R1      ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
3376 022002  012737  017262  003774          MOV     #ER9007,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3377                                        ;+
3378                                        ; Report error with number Initial ERRNRB + 2.
3379                                        ;   "NON-SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE nn AFTER RESET."
3380                                        ;-
3381 022010                                  ERROR                   ;               >>>>> ERROR <<<<<.
     022010  104460                                                                       TRAP    C$ERROR
3382                                        ;+
3383                                        ; End of loop, loop if not all chars have been read from the FIFO.
3384                                        ;-
3385 022012  005203                  10$:    INC     R3              ;SET CODE COUNTER FOR NEXT ITERATION OF LOOP.
3386 022014  020327  000010                  CMP     R3,#NUMLNS      ;TEST FOR ALL CODES READ.
3387 022020  002705                         BLT     2$              ;LOOP IF NOT CHARS READ FROM FIFO.
3388                                        ;+
3389                                        ; Purge the FIFO until DATA.VALID is clear or until too many chars are read.
3390                                        ;-
3391 022022  012704  000022                  MOV     #18.,R4         ;INITIALIZE THE CHARACTER COUNTER.
```

GLOBAL SUBROUTINE - RSTRPT - SEQ 0088

```
3392 022026  010537  003770            MOV     R5,ERRNBR       ;GET INITIAL VALUE OF THE ERROR NUMBER.
3393 022032  062737  000003  003770    ADD     #3,ERRNBR       ;CALCULATE ERROR NUMBER OF NEXT ERROR.
3394 022040  012737  017340  003774    MOV     #ER9008,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3395 022046  017702  160174      12$:  MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE DUT FIFO.
3396 022052  000261                    SEC                     ;INDICATE SUCCESS IN CASE DATA.VALID IS CLEAR.
3397 022054  100051                    BPL     60$             ;EXIT ROUTINE WITH SUCCESS IF DATA.VALID CLEAR.
3398                                 ;+
3399                                 ; We have a character.
3400                                 ; Determine if character is a data character.
3401                                 ;-
3402 022056  012700  070000            MOV     #70000,R0       ;TEST BITS 12 THRU 14 OF THE
3403 022062  040200                    BIC     R2,R0           ;  CODE READ FROM THE DUT FIFO.
3404 022064  001403                    BEQ     14$             ;SKIP THIS ERROR IF CODE IS NOT A DATA CHAR.
3405                                 ;+
3406                                 ; We have an unexpected data character: set up and go to report error.
3407                                 ;-
3408 022066  012701  015573            MOV     #EM9022,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3409 022072  000423                    BR      22$             ;GO TO REPORT THIS ERROR.
3410                                 ;+
3411                                 ; We have an unexpected code.
3412                                 ; Determine if the code is a modem status code.
3413                                 ;-
3414 022074  032702  000001      14$:  BIT     #BIT0,R2        ;TEST MODEM STATUS INDICATOR BIT OF CODE.
3415 022100  001003                    BNE     16$             ;SKIP THIS ERROR IF NOT MODEM STATUS CODE.
3416                                 ;+
3417                                 ; We have a modem status code: set up and go to report error.
3418                                 ;-
3419 022102  012701  015612            MOV     #EM9023,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3420 022106  000415                    BR      22$             ;GO TO REPORT THIS ERROR.
3421                                 ;+
3422                                 ; We have an onboard test code.
3423                                 ; Determine if this code is a BMP code.
3424                                 ;-
3425 022110  032702  000200      16$:  BIT     #BIT7,R2        ;TEST THE ROM VERSION BIT OF THE CODE.
3426 022114  001404                    BEQ     18$             ;GOTO SET UP FOR SELFTEST CODE IF ROM VERSION.
3427 022116  012700  000300            MOV     #300,R0
3428 022122  040200                    BIC     R2,R0           ;TEST THE ERROR TYPE BITS OF THE CODE.
3429 022124  001403                    BEQ     20$             ;SKIP THIS ERROR IF BMP CODE.
3430                                 ;+
3431                                 ; We have a selftest code: set up and go to report error.
3432                                 ;-
3433 022126  012701  015634      18$:  MOV     #EM9024,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3434 022132  000403                    BR      22$             ;GO TO REPORT THIS ERROR.
3435                                 ;+
3436                                 ; We have a BMP code: save it on the queue.
3437                                 ;-
3438 022134  004737  022262      20$:  JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
3439 022140  000401                    BR      24$             ;
3440                                 ;+
3441                                 ; Report the error with error number of Initial ERRNBR + 3.
3442                                 ;    "UNEXPECTED xxx xxxx FOR LINE nn IN FIFO AFTER RESET:"
3443                                 ;-
3444 022142                      22$:  ERROR                   ;               >>>>> ERROR <<<<<.
     022142  104460                                                                TRAP    C$ERROR
3445                                 ;+
3446                                 ; End of loop.
3447                                 ; Count the character we just received, and check for too many received.
```

GLOBAL SUBROUTINE RSTRPT -

```
3448                                        ;-
3449 022144  005304                 24$:     DEC     R4              ;COUNT THIS CHARACTER.
3450 022146  001337                          BNE     12$             ;LOOP IF NOT TOO MANY CHARACTERS PURGED.
3451                                        ;*
3452                                        ; We read too many valid characters while trying to purge the FIFO.
3453                                        ; Report error and exit without success.
3454                                        ;    "FIFO WILL NOT PURGE (DATA.VALID STUCK SET), REMAINDER OF TEST SKIPPED."
3455                                        ;-
3456 022150  012701  015411                  MOV     #EM9017,R1      ;SELECT PROPER ERROR MESSAGE.
3457 022154  010537  003770                  MOV     R5,ERRNBR       ;GET INITIAL ERROR NUMBER.
3458 022160  062737  000004  003770          ADD     #4,ERRNBR       ;CALCULATE INITIAL ERRNBR + 4.
3459 022166  012737  016470  003774          MOV     #ER0503,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3460                                        ;PRINT ERROR REPORT.
3461 022174                                 ERROR                   ;               >>>>> ERROR <<<<<.
     022174  104460                                                                    TRAP    C$ERROR
3462 022176  000241                          CLC                     ;CLEAR THE SUCCESS FLAG.
3463
3464 022200                         60$:     PASS                    ;RESTORE GPRS,
     022200  004736                                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3465 022202  000207                          RTS     PC              ; CARRY   SUCCESS FLAG (SET IF FIFO IS PURGED).
```

GLOBAL SUBROUTINE - RXIE0 -

```
                                .SBTTL  GLOBAL SUBROUTINE                - RXIE0 -
                                ;++ ************************************************************************
                                ;*                          - RECEIVER INTERRUPT DISABLE -
                                ;*      This routine is used to disable receiver interrupts in the DHV11.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      The RX.INT.ENBL bit is cleared in the DUT CSR.
                                ;*               IESTST -contains the updated status of the TX and RX interrupt
                                ;*               enable bits.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,RXIE0
                                ;*
                                ;* COMMENTS:     The contents of the indirect address register field in
                                ;*               the DUT CSR are destroyed.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************
022204  010046                  RXIE0:: MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
022206  106746                          MFPS    -(SP)           ;SAVE PROCESSOR PRIORITY ON STACK.
022210  106427  000340                  MTPS    #PRI07          ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
022214  042737  137777  002330          BIC     #137777,IESTAT  ,CLEAR RX.INT.ENBL BIT IN IESTAT.
022222  013777  002330  160014          MOV     IESTAT,@CSRA    ;DISABLE RX INTERRUPTS.
022230  106426                          MTPS    (SP)+           ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
022232  012600                          MOV     (SP)+,R0        ;RESTORE R0.
022234  000207                         RTS     PC
```

GLOBAL SUBROUTINE - RXIE1 -

```
                                .SBTTL  GLOBAL SUBROUTINE               - RXIE1 -
                                ;++ ************************************************************************
                                ;*                          - RECEIVER INTERRUPT ENABLE -
                                ;*      This routine is used to enable receiver interrupts in the DHV11.
                                ;*
                                ;* INPUTS:      NONE.
                                ;*
                                ;* OUTPUTS:     The RX.INT.ENBL bit is set in the DUT CSR.
                                ;*              IESTST -contains the updated status of the TX and RX interrupt
                                ;*              enable bits.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,RXIE1
                                ;*
                                ;* COMMENTS:    The contents of the indirect address register field in
                                ;*              the DUT CSR are destroyed.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

 022236  052737  000100  002330  RXIE1::  BIS     #BIT06,IESTAT   ;SET RX.INT.ENBL BIT IN IESTAT.
 022244  042737  137677  002330           BIC     #137677,IESTAT  ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
 022252  013777  002330  157764           MOV     IESTAT,@CSRA    ;ENABLE RX INTERRUPTS.
 022260  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                SAVBMP
                        ;** ************************************************************************
                        ;*                      - Save BMP codes Routine -
                        ;*      This routine saves the parameter passed in, onto the BMP code queue
                        ;*      together with the number of the currently executing test.
                        ;*
                        ;* INPUTS:      R2 - Contains the BMP code that is to be placed on the queue.
                        ;*              BMPCQP - Contains address of next location in the bmp queue.
                        ;*              BMPCQB - Label at base of the BMP code queue.
                        ;*              BMPCQE - Label of next location after the end of the BMP queue.
                        ;*              TSTNUM - Contains the number of the current test.
                        ;*
                        ;* OUTPUTS:     BMPCQP -  Incremented by 4.
                        ;*              The contents of the BMP code queue are updated.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,SAVBMP
                        ;*
                        ;* COMMENTS:    If the overflow occurs then the last location will be
                        ;*              overwritten by any subsequent attempts to update the queue.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ************************************************************************

3541 022262             SAVBMP:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
     022262  004537  003776                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3542 022266  013704  002524        MOV     BMPCQP,R4       ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
3543 022272  113724  002326        MOVB    TSTNUM,(R4)+    ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
3544 022276  005204                INC     R4              ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
3545 022300  042702  177400        BIC     #177400,R2      ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
3546 022304  010224                MOV     R2,(R4)+        ;SAVE THE BMP CODE ON THE QUEUE.
3547 022306  020427  002726        CMP     R4,#BMPCQE      ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
3548 022312  103402                BLO     2$              ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
3549 022314  162704  000004        SUB     #4,R4           ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
3550 022320  010437  002524  2$:   MOV     R4,BMPCQP       ;SAVE THE POINTER.
3551
3552 022324             60$:    PASS                     ;RESTORE GPRS.
     022324  004736                         JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3553 022326  000207                RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                SKPSTS -
                        ;** ****************************************************************************
                        ;*                          - Skip Selftest Routine -
                        ;*      This subroutine is used to skip the selftest after a DUT reset has been
                        ;*      initiated.  It must be entered immediately after setting the DUT Master
                        ;*      Reset routine or after the execution of a bus reset (because of timing
                        ;*      considerations).
                        ;*
                        ;* INPUTS:      CSRA - Contains address of the DUT CSR.
                        ;*              TXBFCA - Contains address of DUT DMA Buffer Count register.
                        ;*
                        ;* OUTPUTS:     Skip selftest codes are written to the DUT registers.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,SKPSTS
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: DELAY.
                        ;-- ****************************************************************************

022330                  SKPSTS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
022330  004537  003776                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
022334  012704  000012          MOV     #10.,R4                 ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
022340  004737  020300          JSR     PC,DELAY                ;DELAY FOR 10 MILLI-SECONDS.
                        ;*
                        ; Write skip self test code (52525) to all the indexed DUT Registers.
                        ;-
022344  012701  000050          MOV     #NUMLNS!BIT05,R1 ;FORM IND.ADR.REG FIELD (PLUS M.R. BIT) WORD.
                        ;THE ABOVE INCLUSION OF THE M.R. BIT IS NECESSARY BECAUSE OF THE
                        ;  LACK OF A M.R. BIT WRITE LOCK-OUT ON THE DHV11-M.
022350  012703  052525          MOV     #52525,R3               ;INITIALISE THE SKIP SELF-TEST CODE.
022354  005301          4$:     DEC     R1                      ;SELECT THE NEXT SET OF DEVICE REGISTERS.
022356  013704  002244          MOV     CSRA,R4                 ;GET THE ADDRESS OF THE CSR OF THE DUT.
022362  010124                  MOV     R1,(R4)+                ;SELECT A BANK OF DUT REGISTERS.
022364  010324          6$:     MOV     R3,(R4)+                ;WRITE THE CODE TO A DUT REGISTER.
022366  020437  002262          CMP     R4,TXBFCA               ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
022372  103774                  BLO     6$                      ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
022374  032701  000017          BIT     #17,R1                  ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
022400  001365                  BNE     4$                      ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.

022402                  60$:    PASS                            ;RESTORE GPRS.
022402  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
022404  000207                  RTS     PC
```

```
 3597                                   .SBTTL  GLOBAL SUBROUTINE                - SWAPO -
 3598                                   ;** ****************************************************************************
 3599                                   ;*                        - Swap GPRs With GPR Set 0 Routine -
 3600                                   ;*      This subroutine swaps the present contents of GPRs R1 thru R5 with
 3601                                   ;*      the contents of the number zero GPR save area.  The contents of R0
 3602                                   ;*      are not altered by this subroutine.
 3603                                   ;*
 3604                                   ;* INPUTS:      GPR contents R1 thru R5.
 3605                                   ;*              GPRS0B - Label at base of GPR save area number zero.
 3606                                   ;*
 3607                                   ;* OUTPUTS:     R1 thru R5 contain the previous contents of GPR save area
 3608                                   ;*                      zero words 1 thru 5 respectively.
 3609                                   ;*              GPRS0 - GPR save area 0 words 1 thru 5, contain previous
 3610                                   ;*                      contents of GPRs R1 thru R5 respectively.
 3611                                   ;*
 3612                                   ;* CALLING SEQUENCE:    JSR     PC,SWAPO
 3613                                   ;*
 3614                                   ;* COMMENTS:    The state of the CARRY flag is not alterred by this routine.
 3615                                   ;*
 3616                                   ;* SUBORDINATE ROUTINES CALLED: None.
 3617                                   ;-- ****************************************************************************
 3618
 3619 022406  010046                   SWAPO::  MOV     R0,-(SP)        ;SAVE THE CONTENTS OF R0.
 3620                                   ;+
 3621                                   ; Load the stack from the GPRs.
 3622                                   ;-
 3623 022410  010146                            MOV     R1,-(SP)        ;SAVE THE CONTENTS OF R1.
 3624 022412  010246                            MOV     R2,-(SP)        ;SAVE THE CONTENTS OF R2.
 3625 022414  010346                            MOV     R3,-(SP)        ;SAVE THE CONTENTS OF R3.
 3626 022416  010446                            MOV     R4,-(SP)        ;SAVE THE CONTENTS OF R4.
 3627 022420  010546                            MOV     R5,-(SP)        ;SAVE THE CONTENTS OF R5.
 3628                                   ;+
 3629                                   ; Load the GPRs from the GPR save area 0.
 3630                                   ;-
 3631 022422  012700  002450                    MOV     #GPRS0B,R0      ;GET THE BASE ADDRESS OF GPR SAVE AREA 0.
 3632 022426  012001                            MOV     (R0)+,R1        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 1.
 3633 022430  012002                            MOV     (R0)+,R2        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 2.
 3634 022432  012003                            MOV     (R0)+,R3        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 3.
 3635 022434  012004                            MOV     (R0)+,R4        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 4.
 3636 022436  012005                            MOV     (R0)+,R5        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 5.
 3637                                   ;+
 3638                                   ; Load the GPR save area 0 from the stack.
 3639                                   ;-
 3640 022440  012640                            MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 5 WITH SAVED R5.
 3641 022442  012640                            MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 4 WITH SAVED R4.
 3642 022444  012640                            MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 3 WITH SAVED R3.
 3643 022446  012640                            MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 2 WITH SAVED R2.
 3644 022450  012640                            MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 1 WITH SAVED R1.
 3645
 3646 022452  012600                            MOV     (SP)+,R0        ;RESTORE THE INITIAL VALUE OF R0.
 3647
 3648 022454  000207                           RTS     PC
```

GLOBAL SUBROUTINE TSABRT -

```
                                .SBTTL  GLOBAL SUBROUTINE               - TSABRT -
                                ;++ ************************************************************************
                                ;*                      - TEST ABORT ROUTINE -
                                ;*      This subroutine is used when a non-test related error has been found
                                ;*      during the execution of the current test.
                                ;*      It is used to inform the operator that the current test has been
                                ;*      aborted.
                                ;*
                                ;* INPUTS:      ERRMSG - Contains the name of the current test.
                                ;*              ERRNBR - Contains the correct error number.
                                ;*              The remainder of the ERRTBL is correctly initialised.
                                ;*
                                ;* OUTPUTS:     Messages are reported to the operator.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TSABRT
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: ER1603.
                                ;-- ************************************************************************

022456                          TSABRT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
022456  004537  003776                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022462  012701  022500                  MOV     @2$,R1          ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
022466  012737  017010  003774          MOV     @ER1603,ERRBLK  ;SET-UP THE ERROR REPORTING ROUTINE.
022474                                  ERROR                   ;               >>>>> ERROR <<<<<.
022474  104460                                                                          TRAP    C$ERROR
022476  000432                          BR      60$             ;
022500     040     116     117  2$:     .ASCIZ / NON-RELATED TEST ERROR FOUND DURING TEST EXECUTION/
022503     116     055     122
022506     105     114     101
022511     124     105     104
022514     040     124     105
022517     123     124     040
022522     105     122     122
022525     117     122     040
022530     106     117     125
022533     116     104     040
022536     104     125     122
022541     111     116     107
022544     040     124     105
022547     123     124     040
022552     105     130     105
022555     103     125     124
022560     111     117     116
022563     000
                                .EVEN
022564                  60$:    PASS                    ;RESTORE GPRS.
022564  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
022566  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXDSBL -
                        ;** ************************************************************************
                        ;*                        - Transmitter Disable -
                        ;*      This subroutine is used to disable transmission on selected lines by,
                        ;*      clearing the associated TX.ENABLE bit on the DUT.
                        ;*
                        ;* INPUTS:      R5 - Bit's set correspond to lines on which to clear TX.ENABLE.
                        ;*              CSRA   Contains the address of the DUT CSR.
                        ;*              IESTAT - Contains the state of TXIE and RXIE bits in the CSR.
                        ;*              NUMLNS - Equated to be the maximum number of lines availabie.
                        ;*              TXAD2A - Contains the address of the TBUFFAD2 register.
                        ;*
                        ;* OUTPUTS:     R5 - Bit's set indicate the initial states of all TX.ENBLE bits.
                        ;*              TBUFFAD2 - The state of the TX.ENBLE bit may be altered.
                        ;*              The contents of the IND.ADD.REG field in the CSR are destroyed.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TXDSBL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ; - ************************************************************************

022570                  TXDSBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
022570  004537  003776                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
022574  010500          MOV     R5,R0           ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
022576  012701  000001  MOV     #BIT0,R1        ;INITIALIZE THE SELECTED LINE BIT MASK.
022602  013702  002260  MOV     TXAD2A,R2       ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
022606  005202          INC     R2              ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
022610  012703  000010  MOV     #NUMLNS,R3      ;GET MAXIMUM LINE NUMBER PLUS ONE.
022614  013704  002330  MOV     IESTAT,R4       ;GET THE STATES OF THE INT ENABLE BITS.
022620  005005          CLR     R5              ;LOG POSSIBLE TX DISABLED ON ALL LINES.
                        ;+
                        ; Select every line in turn, and log the state of each TX.ENABLE bit.
                        ;-
022622  010477  157416  2$:     MOV     R4,@CSRA        ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
022626  105712          TSTB    (R2)            ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
022630  100001          BPL     4$              ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
022632  050105          BIS     R1,R5           ;LOG TX ENABLE BIT SET FOR SELECTED LINE.
                        ;+
                        ;  Clear TX.ENBLE on lines that have a corresponding bit set in the tx disable
                        ;  line bit map.
                        ;-
022634  030100          4$:     BIT     R1,R0           ;CHECK STATE OF DISABLE LINE BIT MAP.
022636  001402          BEQ     6$              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
022640  142712  000200  BICB    #BIT7,(R2)      ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
022644  005204          6$:     INC     R4              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
022646  006301          ASL     R1              ;SHIFT BIT MAP FOR NEXT LINE.
022650  005303          DEC     R3              ;DECREMENT LINE NUMBER.
022652  001363          BNE     2$              ;LOOP TO CHECK NEXT LINE.

022654                  60$:    PASS    R5              ;RESTORE GPRS,EXCEPT
022654  010566  000014                          MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
022660  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
                                                ;R5   PREVIOUS STATES OF ALL TX.ENABLE BITS.
022662  000207          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXENBL -
                        ;++ ************************************************************************
                        ;*                      - Transmitter Enable -
                        ;*      This subroutine is used to enable transmission on selected lines by
                        ;*      setting the associated TX.ENABLE bit on the DUT.
                        ;*
                        ;* INPUTS:      R5 - Bit's set correspond to lines on which to set TX.ENABLE.
                        ;*              CSRA - Contains the address of the DUT CSR.
                        ;*              IESTAT - Contains the state of TXIE and RXIE bits in the CSR.
                        ;*              NUMLNS - Equated to be the maximum number of lines available.
                        ;*              TXAD2A - Contains the address of the TBUFFAD2 register.
                        ;*
                        ;* OUTPUTS:     R5 - Bit's set indicate previously disabled lines.
                        ;*              TBUFFAD2 - The state of the TX.ENBLE bit may be altered.
                        ;*              The contents of the IND.ADD.REG field in the CSR are destroyed.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TXENBL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

022664                  TXENBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
022664  004537  003776                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022670  010500          MOV     R5,R0           ;COPY BIT MAP OF LINES TO ENABLE.
022672  012701  000001  MOV     #BIT0,R1        ;INITIALIZE THE SELECTED LINE BIT MASK.
022676  013702  002260  MOV     TXAD2A,R2       ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
022702  005202          INC     R2              ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
022704  012703  000010  MOV     #NUMLNS,R3      ;GET MAXIMUM LINE NUMBER.
022710  013704  002330  MOV     IESTAT,R4       ;GET THE STATES OF THE INT ENABLE BITS.
022714  005005          CLR     R5              ;CLEAR TX.ENABLE BIT LOG OF DISABLED LINES.
                        ;+
                        ; Select every line in turn,and log any TX.ENBLE bit that is clear.
                        ;-
022716  010477  157322  2$:     MOV     R4,@CSRA        ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
022722  105712          TSTB    (R2)            ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
022724  100401          BMI     4$              ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
022726  050105          BIS     R1,R5           ;LOG TX ENABLE BIT CLEAR FOR SELECTED LINE.
                        ;+
                        ;  Set TX.ENBLE on lines that have a corresponding bit set in the tx enable
                        ;  line bit map.
                        ;-
022730  030100          4$:     BIT     R1,R0           ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
022732  001402          BEQ     6$              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
022734  152712  000200  BISB    #BIT7,(R2)      ;ENABLE TRANSMISSION ON SELECTED LINE.
022740  005204          6$:     INC     R4              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
022742  006301          ASL     R1              ;SHIFT BIT MAP FOR NEXT LINE.
022744  005303          DEC     R3              ;DECREMENT LINE NUMBER.
022746  001363          BNE     2$              ;LOOP TO CHECK NEXT LINE.

022750                  60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
022750  010566  000014                  MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
022754  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;R5 - LINE BIT MAP CORRESPONDING TO THE
                                                                ; PREVIOUS LINES THAT WERE DISABLED.
022756  000207          RTS     PC
```

GLOBAL SUBROUTINE - TXIE0 -

```
                                .SBTTL  GLOBAL SUBROUTINE               - TXIE0 -
                                ;++ ************************************************************************
                                ;*                          - TRANSMITTER INTERRUPT DISABLE -
                                ;*      This routine is used to disable transmitter interrupts in the DHV11.
                                ;*
                                ;* INPUTS:      NONE.
                                ;*
                                ;* OUTPUTS:     The TX.INT.ENBL bit is cleared in the DUT CSR.
                                ;*              IESTST -contains the updated status of the TX and RX interrupt
                                ;*              enable bits.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TXIE0
                                ;*
                                ;* COMMENTS:    The contents of the indirect address register field in
                                ;*              the DUT CSR are destroyed.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************
022760  010046                  TXIE0::  MOV    R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
022762  106746                         MFPS    -(SP)           ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
022764  106427  000340                 MTPS    @PRI07          ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
022770  042737  177677  002330          BIC    @177677,IESTAT  ;CLEAR TX.INT.ENBL BIT IN IESTAT.
022776  013777  002330  157240          MOV    IESTAT,@CSRA    ;DISABLE TX INTERRUPTS.
023004  106426                          MTPS   (SP)+           ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
023006  012600                          MOV    (SP)+,R0        ;RESTORE R0.
023010  000207                         RTS     PC
```

GLOBAL SUBROUTINE - TXIE1 -

```
                                .SBTTL  GLOBAL SUBROUTINE               - TXIE1 -
                                ;++ ******************************************************************************
                                ;*                          - TRANSMITTER INTERRUPT ENABLE -
                                ;*      This routine is used to enable transmitter interrupts in the DHV11.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      The TX.INT.ENBL bit is set in the DUT CSR.
                                ;*               IESTST -contains the updated status of the TX and RX interrupt
                                ;*               enable bits.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TXIE1
                                ;*
                                ;* COMMENTS:     The contents of the indirect address register field in
                                ;*               the DUT CSR are destroyed.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ******************************************************************************

023012  052737  040000  002330  TXIE1::  BIS     #BIT14,IESTAT   ;SET TX.INT.ENBL BIT IN IESTAT.
023020  042737  137677  002330          BIC     #137677,IESTAT  ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
023026  013777  002330  157210          MOV     IESTAT,@CSRA    ;DISABLE TX INTERRUPTS.
023034  000207                         RTS     PC
```

```
                                        .SBTTL  GLOBAL SUBROUTINE                - UNSDIV -
                                        ;++ ************************************************************************
                                        ;*                        - Unsigned Divide Routine -
                                        ;*      This subroutine is used to divide a 32 bit unsigned dividend by a
                                        ;*      16 bit unsigned divisor giving a 16 bit quotient.  All numbers are
                                        ;*      considered to be unsigned.  A success flag is not set on return if
                                        ;*      the quotient was too big to be contained in 16 bits.
                                        ;*
                                        ;* INPUTS:      R1   The divisor, unsigned, 16 bits.
                                        ;*              R2 - Most significant word of the dividend, unsigned, 16 bits.
                                        ;*              R3 - Least significant word of the dividend, unsigned, 16 bits.
                                        ;*
                                        ;* OUTPUTS:     R1 - Quotient, unsigned, 16 bits (177777 if overflow).
                                        ;*              CARRY - Success flag, set if complete quotient fits in 16 bits.
                                        ;*
                                        ;* CALLING SEQUENCE:    JSR     PC,UNSDIV
                                        ;*
                                        ;* COMMENTS:    If the divisor is 0 the quotient is returned as all ones
                                        ;*              (177777) and the carry is clear regardless of the dividend.
                                        ;*
                                        ;* SUBORDINATE ROUTINES CALLED: None.
                                        ;-- ************************************************************************

023036                                  UNSDIV:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023036  004537  003776                                 JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
                                        ;+
                                        ; Check for quotient greater than 16 bits condition.
                                        ;-
023042  010204                                  MOV     R2,R4                   ;GET MSW OF DIVIDEND FOR SUBTRACT.
023044  160104                                  SUB     R1,R4                   ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
023046  103403                                  BCS     2$                      ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
023050  012701  177777                          MOV     #-1,R1                  ;SET QUOTIENT TO ALL ONES (177777),
023054  000442                                  BR      60$                     ;EXIT WITH CARRY CLEAR.
                                        ;+
                                        ; Set up counters and various working GPRs.
                                        ;-
023056  005004                          2$:     CLR     R4                      ;CLEAR THE LSW OF THE DIVISOR.
023060  000241                                  CLC                             ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR
023062  006001                                  ROR     R1                      ; DIVISOR BY
023064  006004                                  ROR     R4                      ; 2(UNSIGNED)
023066  012700  000020                          MOV     #16.,R0                 ;SET UP INITIAL SHIFT COUNT TO 16.
                                        ;+
                                        ; The subtract and shift loop.
                                        ;
023072  010246                          4$:     MOV     R2, (SP)                ;SAVE MSWORD OF DIVIDEND.
023074  010346                                  MOV     R3,-(SP)                ;SAVE LSWORD OF DIVIDEND.
023076  160403                                  SUB     R4,R3                   ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
023100  005602                                  SBC     R2                      ;MSWORD DIVIDEND - BORROW .
023102  103402                                  BCS     6$                      ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
023104  160102                                  SUB     R1,R2                   ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
023106  103003                                  BCC     8$                      ;IF NO BORROW, IT WENT, CARRY IS CLEAR.
                                            ;+
                                            ; It didn't go, so we shift a 1 into the quotient (complemented later).
                                            ; Carry is set.
                                            ;-
023110  012603                          6$:     MOV     (SP)+,R3                ;RESTORE LSWORD OF DIVIDEND.
023112  012602                                  MOV     (SP)+,R2                ;RESTORE MSWORD OF DIVIDEND.
```

GLOBAL SUBROUTINE - UNSDIV -

```
3897 023114  000401                    BR      10$             ;GOTO SHIFT 1 INTO THE QUOTIENT.
3898                                ;+
3899                                ; It went, so we restore the stack and shift a 0 into quotient (will be
3900                                ; complemented later).   Carry is clear.
3901                                ;-
3902 023116  012626              8$:   MOV     (SP)+,(SP)+     ;POP THE SAVED DIVIDEND OFF OF THE STACK.
3903                                ;+
3904                                ; Shift the result of the subtract attempt into the quotient shift reg.
3905                                ;-
3906 023120  006105             10$:   ROL     R5              ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
3907 023122  000241                    CLC                     ;DIVIDE THE
3908 023124  006001                    ROR     R1              ;  DEVISOR BY
3909 023126  006004                    ROR     R4              ;  2 (UNSIGNED).
3910 023130  005300                    DEC     R0              ;COUNT THIS SHIFT AND SUBTRACT.
3911 023132  001357                    BNE     4$              ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
3912 023134  005105                    COM     R5              ;GET QUOTIENT FROM INVERTED QUOTIENT.
3913                                ;+
3914                                ; Now we either round up or leave quotient alone.
3915                                ;-
3916 023136  000241                    CLC                     ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
3917 023140  006103                    ROL     R3              ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
3918 023142  103402                    BCS     12$             ;IF CARRY FROM SHIFT, ROUND UP.
3919 023144  160403                    SUB     R4,R3           ;SUBTRACT DIVISOR FROM DIVIDEND.
3920 023146  103403                    BCS     14$             ;IF BORROW, DON'T ROUND UP.
3921                                ;+
3922                                ; Round up, extra subtract went.
3923                                ;
3924 023150  005205             12$:   INC     R5              ;INCREMENT THE QUOTIENT BY ONE.
3925 023152  001001                    BNE     14$             ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
3926 023154  005305                    DEC     R5              ;DON'T LET ROUNDING CAUSE OVERFLOW.
3927                                ;+
3928                                ; All done, pass quotient and exit.
3929                                ;
3930 023156  010501             14$:   MOV     R5,R1           ;PASS QUOTIENT BACK IN R1.
3931 023160  000261                    SEC                     ;INDICATE NO OVERFLOW.
3932
3933 023162                     60$:   PASS    R1              ;RESTORE GPRS, LFAVE THE FOLLOWING INTACT:
     023162  010166  000004                        MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
     023166  004736                                JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3934                                                         ;R1 - 16 BIT, UNSIGNED QUOTIENT,
3935 023170  000207                    RTS     PC              ;CARRY - SET INDICATES NO OVERFLOW (SUCCESS).
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - WAIBIC -
                        ;** ************************************************************************
                        ;*                      - Wait For Bit Clear Routine -
                        ;*      This subroutine waits for the specified bit to become clear.  If the
                        ;*      specified bit goes to a clear state within the specified time-out
                        ;*      period a success indication is returned by this routine.
                        ;*      The last value which is read looking for the condition is returned to
                        ;*      allow the use of this routine to look for destructive read conditions.
                        ;*
                        ;* INPUTS:      R1 - Time-out value and bit number indication:
                        ;*                Bits 15 thru 12 - Number of bit to test (range 0 thru 15).
                        ;*                Bits 11 thru 0  - Time-out value  n milli-seconds (4095 max).
                        ;*              R2   Address of word containing the bit to test.
                        ;*                   MSLCNT.
                        ;*
                        ;* OUTPUTS:     R2   The last word which was read to check for the condition.
                        ;*              CARRY - Success flag (CARRY set if bit clr before time-out).
                        ;*
                        ;* CALLING SEQUENCE:    MOV     #130040,R1      ;PASS BIT 11 (13 OCTAL) AND
                        ;*                                              ;  32 (40 OCTAL) MS DELAY.
                        ;*                       MOV     #LABEL,R2       ;TEST BIT IN WORD AT "LABEL".
                        ;*                       JSR     PC,WAIBIC       ;WAIT 32 MS FOR BIT 11 TO CLR.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: MSLGET.
                        ;   ************************************************************************

023172                  WAIBIC:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023172  004537  003776                       JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023176  010204                  MOV     R2,R4           ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
023200  010102                  MOV     R1,R2
023202  042701  170000          BIC     #170000,R1      ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
023206  042702  007777          BIC     #7777,R2        ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
023212  000302                  SWAB    R2              ;PUT LINE NUMBER FIELD IN LSBYTE.
023214  006202                  ASR     R2              ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
023216  006202                  ASR     R2              ;  POSITION TO USE IT AS A WORD TABLE OFFSET
023220  006202                  ASR     R2              ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
023222  016202  002410          MOV     BITTBL(R2),R2   ;GET BIT MAP OF LINE TO TEST FROM TABLE.
023226  005003                  CLR     R3              ;INDICATE THAT THE BIT SHOULD BE CLR.
023230  004737  020340          JSR     PC,MSLGET       ;WAIT FOR THE BIT TO BE CLR WITHIN TIME-OUT.
                                                        ;  CARRY IS CORRECT UPON MSLGET RETURN.
023234  010002                  MOV     R0,R2           ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
023236                  60$:    PASS    R2              ;RESTORE GPRS, EXCEPT THE FOLLOWING:
023236  010266  000006                       MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
023242  004736                           JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                        ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
023244  000207                  RTS     PC              ;  CARRY   SUCCESS FLAG (SET IF BIT FOUND CLR).
```

GLOBAL SUBROUTINE - WAIBIS -

```
                                .SBTTL  GLOBAL SUBROUTINE               - WAIBIS -
                                ;++ ************************************************************************
                                ;*                      - Wait For Bit Set Routine -
                                ;*      This subroutine waits for the specified bit to become set.  If the
                                ;*      specified bit goes to a set state within the specified time-out
                                ;*      period a success indication is returned by this routine.
                                ;*      The last value which is read looking for the condition is returned to
                                ;*      allow the use of this routine to look for destructive read conditions.
                                ;*
                                ;* INPUTS:      R1 - Time-out value and bit number indication:
                                ;*                Bits 15 thru 12 - Number of bit to test (range 0 thru 15).
                                ;*                Bits 11 thru 0  - Time-out value in milli-seconds (4095 max).
                                ;*              R2 - Address of word containing the bit to test.
                                ;*                   MSLCNT.
                                ;*
                                ;* OUTPUTS:     R2 - The last word which was read to check for the condition.
                                ;*              CARRY - Success flag (CARRY set if bit set before time-out).
                                ;*
                                ;* CALLING SEQUENCE:    MOV     #130040,R1      ;PASS BIT 11 (13 OCTAL) AND
                                ;*                                              ;  32 (40 OCTAL) MS DELAY.
                                ;*                      MOV     #LABEL,R2       ;TEST BIT IN WORD AT "LABEL".
                                ;*                      JSR     PC,WAIBIS       ;WAIT 32 MS FOR BIT 11 TO SET.
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: MSLGET.
                                ;-- ************************************************************************

023246                          WAIBIS:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
023246  004537  003776                   JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
023252  010204                  MOV     R2,R4           ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
023254  010102                  MOV     R1,R2
023256  042701  170000          BIC     #170000,R1      ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
023262  042702  007777          BIC     #7777,R2        ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
023266  000302                  SWAB    R2              ;PUT LINE NUMBER FIELD IN LSBYTE.
023270  006202                  ASR     R2              ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
023272  006202                  ASR     R2              ;  POSITION TO USE IT AS A WORD TABLE OFFSET
023274  006202                  ASR     R2              ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
023276  016202  002410          MOV     BITTBL(R2),R2   ;GET BIT MAP OF LINE TO TEST FROM TABLE.
023302  010203                  MOV     R2,R3           ;INDICATE THAT THE BIT SHOULD BE SET.
023304  004737  020340          JSR     PC,MSLGET       ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
                                                        ;  CARRY IS CORRECT UPON MSLGET RETURN.
023310  010002                  MOV     R0,R2           ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
023312                  60$:    PASS    R2              ;RESTORE GPRS, EXCEPT THE FOLLOWING:
023312  010266  000006                          MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
023316  004736                              JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
                                                        ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
023320  000207                  RTS     PC              ;  CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).
```

GLOBAL SUBROUTINE - WDPDR -

```
4029                                    .SBTTL  GLOBAL SUBROUTINE               - WDPDR -
4030                                    ;++ ************************************************************************
4031                                    ;*                      - Write Data Pattern to Device Registers -
4032                                    ;*      This routine writes a rotated data pattern to each of the 6 device
4033                                    ;*      registers of each active line of the device under test.
4034                                    ;*      The data pattern is rotated once after each write to a device register
4035                                    ;*              on a particular line.  The starting data pattern for each line
4036                                    ;*              is rotated once after writing all the registers on a particular
4037                                    ;*              line.  This leads to the following data pattern:
4038                                    ;*                      Line 0, register 0 - shifted 0 bit positions
4039                                    ;*                      Line 0, register 1 - shifted 1 bit position
4040                                    ;*                      ...
4041                                    ;*                      Line 1, register 0 - shifted 1 bit position
4042                                    ;*                      Line 2, register 1 - shifted 2 bit positions
4043                                    ;*                      ...
4044                                    ;*      Any bits fields in the device registers that cannot be altered
4045                                    ;*      are masked out of the data pattern before it is written.
4046                                    ;*      This routine will use either MOV, MOVB, BIS, BISB, BIC, or BICB
4047                                    ;*      instructions.  The upper or lower byte can be specified for writing.
4048                                    ;*
4049                                    ;* INPUTS:      R2 - Used to pass in the data pattern to be rotated & written.
4050                                    ;*              R3 - Byte indicator (- => lo byte, + => hi byte, 0 => both).
4051                                    ;*              R4 - Operation type indicator (- => BIC, + => BIS, 0 => MOV).
4052                                    ;*              ACTLNS - Bit map of the active lines on the device under test.
4053                                    ;*              CSRA - Contains the CSR address of the Device under test.
4054                                    ;*              DRADRT - Base address of device register address table.
4055                                    ;*              LPR0 - Equated to LPR reg offset from device CSR address.
4056                                    ;*              NUMLNS - Number of lines on the device under test.
4057                                    ;*              TXBFC0 - Equated to TBUFFCT reg offset from device CSR address.
4058                                    ;*              UNBTTB - Base address of the unused bit table.
4059                                    ;*
4060                                    ;* OUTPUTS:     Device registers on all active device lines are modified.
4061                                    ;*
4062                                    ;* CALLING SEQUENCE:    JSR     PC,WDPDR
4063                                    ;*
4064                                    ;* COMMENTS:    This routine does not write any data to the TX.CHAR registers.
4065                                    ;*              The CSR is cleared except for the IND.ADR.REG field.
4066                                    ;*
4067                                    ;* SUBORDINATE ROUTINES CALLED: ROLDAP.
4068                                    ;-- ************************************************************************
4069
4070 023322                             WDPDR:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
    023322  004537  003776                              JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
4071                                    ;*
4072                                    ; Set up outer loop which writes the data pattern to each line's registers
4073                                    ;
4074 023326  005005                             CLR     R5                      ;CLEAR LINE COUNTER TO SELECT LINE 0.
4075                                    ;*
4076                                    ; The outer loop follows.  Each pass through this loop writes data to all of
4077                                    ; the device registers for a particular line if it is active
4078                                    ;
4079 023330  010204                     2$:     MOV     R2,R4                   ;SAVE THE OUTER LOOP DATA PATTERN.
4080 023332  010577  156706                     MOV     R5,@CSRA                ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
4081 023336  006305                             ASL     R5                      ;TURN LINE NUMBER INTO A WORD OFFSET.
4082 023340  036537  002410  002236              BIT     @BITTBL(R5),ACTLNS
4083 023346  001451                             BEQ     20$                     ;LINE ACTIVE?  NO, SKIP THIS LINE.
4084 023350  012701  000004                     MOV     #LPR0,R1                ;YES, INITIALIZE THE REGISTER OFFSET
```

```
4085                                    ;+
4086                                    ;       The inner loop follows.  Each pass through this loop writes data to a
4087                                    ;       device register.
4088                                    ;-
4089 023354  010200                     4$:      MOV     R2,R0
4090 023356  046100  002264                      BIC     UNBTTB(R1),R0   ;CLEAR BIT FIELDS FOR UNUSED REGISTER BITS.
4091 023362  016103  002244                      MOV     DRADRT(R1),R3   ;GET THE ADDRESS OF THE DEVICE REGISTER.
4092 023366  005766  000010                      TST     R3SLOT(SP)      ;CHECK THE OPERAND TYPE INDICATOR.
4093 023372  003402                             BLE     6$              ;HIGH BYTE?  NO, SKIP HIGH BYTE ADDRESS SET UP.
4094 023374  005203                              INC     R3              ;YES, SET THE REG ADDRESS TO THE HIGH BYTE.
4095 023376  000300                              SWAB    R0              ;MOVE HIGH BYTE DATA INTO THE LOW BYTE.
4096 023400  005766  000010             6$:      TST     R3SLOT(SP)      ;CHECK THE OPERAND TYPE INDICATOR.
4097 023404  001412                             BEQ     12$             ;WORD ACCESS?  YES, GO PERFORM WORD ACCESS.
4098                                        ;+
4099                                        ;Perform byte access to the specified byte of the specified register.
4100                                        ;-
4101 023406  005766  000012                      TST     R4SLOT(SP)      ;NO, CHECK THE ACCESS TYPE INDICATOR.
4102 023412  100403                             BMI     8$              ;USE BIC?  YES, GO PERFORM BICB INSTRUCTION.
4103 023414  001404                             BEQ     10$             ;USE MOV?  YES, GO PERFORM MOVB INSTRUCTION.
4104 023416  150013                              BISB    R0,(R3)         ;NEITHER.  PERFORM BISB ACCESS TO REGISTER.
4105 023420  000415                             BR      18$
4106 023422  140013                     8$:      BICB    R0,(R3)         ;PERFORM BICB ACCESS TO REGISTER.
4107 023424  000413                             BR      18$
4108 023426  110013                     10$:     MOVB    R0,(R3)         ;PERFORM MOVB ACCESS TO REGISTER.
4109 023430  000411                             BR      18$
4110                                        ;+
4111                                        ;Perform word access to the specified register.
4112                                        ;-
4113 023432  005766  000012             12$:     TST     R4SLOT(SP)      ;CHECK THE ACCESS TYPE INDICATOR.
4114 023436  100403                             BMI     14$             ;USE BIC?  YES, GO PERFORM BIC INSTRUCTION.
4115 023440  001404                             BEQ     16$             ;USE MOV?  YES, GO PERFORM MOV INSTRUCTION.
4116 023442  050013                              BIS     R0,(R3)         ;NEITHER.  PERFORM BIS ACCESS TO REGISTER.
4117 023444  000403                             BR      18$
4118 023446  040013                     14$:     BIC     R0,(R3)         ;PERFORM BIC ACCESS TO REGISTER.
4119 023450  000401                             BR      18$
4120 023452  010013                     16$:     MOV     R0,(R3)         ;PERFORM MOV ACCESS TO REGISTER.
4121                                        ;+
4122                                        ; Prepare the data pattern and offset for the next register on th's line.
4123                                        ;-
4124 023454  004737  021570             18$:     JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
4125 023460  062701  000002                      ADD     #2,R1           ;INCREMENT OFFSET FOR NEXT REGISTER.
4126 023464  020127  000016                      CMP     R1,#TXBFCO      ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
4127 023470  003731                             BLE     4$              ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
4128                                        ;+
4129                                        ; Back into the outer loop.  Now set up for next line.  Loop if not done.
4130                                        ;
4131 023472  010402                     20$:     MOV     R4,R2           ;SET UP TO ROTATE THE DATA PATTERN.
4132 023474  004737  021570                      JSR     PC,ROLDAP       ;ROTATE THE DATA PATTERN.
4133 023500  006205                              ASR     R5              ;CONVERT BACK TO LINE NUMBER FROM WORD OFFSET.
4134 023502  005205                              INC     R5              ;COUNT THIS LINE.
4135 023504  020527  000010                      CMP     R5,#NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
4136 023510  002707                              BLT     2$              ;LOOP IF SOME LINES NOT DONE.
4137
4138 023512                             60$:     PASS                    ;RESTORE GPRS.
     023512  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
4139 023514  000207                             RTS     PC
```

GLOBAL SUBROUTINE WTWLNC -

```
                        .SBTTL  GLOBAL SUBROUTINE                - WTWLNC -
                        ;** ************************************************************************
                        ;*                      - Line Control Register Setup Routine -
                        ;*      This subroutine is used to set the Device Under Test (DUT) Line
                        ;*      Control Registers (LNCTRL) to the specified state.  Only the LNCTRLS
                        ;*      for the specified lines are altered.
                        ;*
                        ;* INPUTS:      R0 - New line parameters.
                        ;*              R5 - Bit map of lines to be altered.
                        ;*              CSRA - Contains address of the DUT CSR.
                        ;*              IESTAT - Contains the current state of the TX and RX interrupt
                        ;*               enable b'ts in the CSR.
                        ;*              LNCTRA - Contains address of the DUT LNCTRL registers.
                        ;*
                        ;* OUTPUTS:     LNCTRL   Specified DUT Line Control Registers are altered.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WTWLNC
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
                        ;- ************************************************************************

023516                  WTWLNC:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023516  004537  003776                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; Set up the parameters for the call to ALTFLD
                        ;
023522  013701  002254          MOV     LNCTRA,R1       ;SET UP THE REGISTER ADDRESS PARAMETER.
023526  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
023530  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
023532  012704  177777          MOV     # 1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED
                        ;*
                        ; Call the subroutine which alters the register contents.
                        ;-
023536  004737  017630          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

023542                  60$:    PASS                    ;RESTORE GPRS.
023542  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023544  000207          RTS     PC
```

GLOBAL SUBROUTINE WTWLNS -

```
                        .SBTTL  GLOBAL SUBROUTINE                - WTWLNS -
                        ;*****************************************************************************
                        ;*                      - Write Word to all Lines routine -
                        ;*      This subroutine writes a specified word to the specified DHV device
                        ;*      register for each of the DHV lines.  It could be used to clear all
                        ;*      of the LNCTRL registers or to initialize all of the LPR registers to
                        ;*      the same parameters.
                        ;*
                        ;* INPUTS:      R1 - Address of the specified registers.
                        ;*              R2 - Word to write into the specified registers.
                        ;*              IESTAT   Saved states of the TX.IE and RX.IE bits.
                        ;*              MAPLNS - Equated to bit map of lines on device (8 for DHV11).
                        ;*                CSRA.
                        ;*
                        ;* OUTPUTS:     DEVICE REGISTERS   Specified registers giver new value.
                        ;*              CSR IND.ADR.REG field - Destroyed.
                        ;*              CSR Interrupt Enable bits - Set to states in IESTAT.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WTWLNS
                        ;*
                        ;* COMMENTS:    Note that the specified registers for all lines are altered
                        ;*              by this routine.  This routine should NOT be used to alter
                        ;*              the states of partial register fields or to alter a register
                        ;*              for fewer than all of the lines.
                        ;*              The specified registers are read before being written.
                        ;*
                        ;* SUBROUTINES CALLED: ALTFLD.
                        ;*****************************************************************************
                        
023546                  WTWLNS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023546  004537  003776                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; Set up the bit map of lines to change and mask of bits to alter parameters.
                        ;
023552  012703  000377          MOV     @MAPLNS,R3      ;GET THE BIT MAP OF LINES TO CHANGE.
023556  012704  177777          MOV     @-1,R4          ;INDICATE ALL 16 BITS TO BE CHANGED.
                        ;*
                        ; Call the subroutine to write the specified registers.
                        ;
023562  004737  017630          JSR     PC,ALTFLD       ;CHANGE THE REGISTERS.

023566                  60$:    PASS                    ;RESTORE GPRS.
023566  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023570  000207          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - WTWLPR -
                        ;** ****************************************************************************
                        ;*                        - Line Parameter Register Setup Routine -
                        ;*      This subroutine is used to set the Device Under Test (DUT) Line
                        ;*      Parameter Registers (LPR) to the specified state.  Only the LPRs for
                        ;*      the specified lines are altered.
                        ;*
                        ;* INPUTS:      R0 - New line parameters.
                        ;*              R5 - Bit map of lines to be altered.
                        ;*              CSRA - Contains address of the DUT CSR.
                        ;*              IESTAT - Contains the current state of the TX and RX interrupt
                        ;*               enable bits in the CSR.
                        ;*              LPRA - Contains address of the DUT LPR.
                        ;*
                        ;* OUTPUTS:     LPR - Specified DUT Line Paramter Registers are altered.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WTWLPR
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
                        ; - ****************************************************************************

023572                  WTWLPR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023572  004537  003776                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; Set up the parameters for the call to ALTFLD.
                        ;
023576  013701  002250          MOV     LPRA,R1         ;SET UP THE REGISTER ADDRESS PARAMETER.
023602  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
023604  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
023606  012704  177777          MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                        ;*
                        ; Call the subroutine which alters the register contents.
                        ;-
023612  004737  017630          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

023616                  60$:    PASS                            ;RESTORE GPRS.
023616  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023620  000207          RTS     PC
```

```
                        .SBTTL  INTERRUPT SERVICE ROUTINE        CACHRX -
                        ;++ ******************************************************************************
                        ;*      - Catch Receiver interrupt.
                        ;*      This routine is used in several tests,to log a count of the
                        ;*      number of receiver interupts that occur.
                        ;*
                        ;* INPUTS:      CSRA - Contains the address of the CSR.
                        ;*              RXINTC - Holds the count of the number of RX interrupts
                        ;*                       that occurred.
                        ;*
                        ;* OUTPUTS:     RXINTC - Contains the updated interrupt count.
                        ;*
                        ;*
                        ;* CALLING SEQUENCE:    Put the address of the label CACHRX in the vector
                        ;*                      location.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: none
                        ;-- ******************************************************************************

023622                  CACHRX::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
023622  004537  003776                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023626  013701  002334          MOV     RXINTC,R1       ;GET THE RECEIVER INTERRUPT COUNT
023632  005201              INC     R1              ;INCREMENT THE COUNT
023634  102001              BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED
023636  005301              DEC     R1              ;RESET THE COUNT TO 177777
023640  010137  002334      2$:     MOV     R1,RXINTC       ;SAVE NEW COUNT VALUE
023644                  60$:    PASS                    ;RESTORE GPRS.
023644  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023646  000002              RTI
```

```
                                        .SBTTL  INTERRUPT SERVICE ROUTINE       - CACHTX -
                                        ;++ ************************************************************************
                                        ;*      - Catch Transmiter interrupt.
                                        ;*      This routine is used in several tests,to log a count of the
                                        ;*      number of transmission interrupts that occur.
                                        ;*
                                        ;* INPUTS:      CSRA - Contains the address of the CSR.
                                        ;*              TXINTC - Holds the count of the number of TX interrupts
                                        ;*                       that occurred.
                                        ;*
                                        ;* OUTPUTS:     TXINTC - Contains the updated interrupt count.
                                        ;*
                                        ;*
                                        ;* CALLING SEQUENCE:    Put the address of the label CACHTX in the vector
                                        ;*                      location.
                                        ;*
                                        ;* COMMEN S:
                                        ;*
                                        ;* SUBORDINATE ROUTINES CALLED: none
                                        ;-- ************************************************************************

023650                                  CACHTX::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
023650  004537  003776                                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023654  013701  002340                          MOV     TXINTC,R1       ;GET THE TRANSMISSION INTERRUPT COUNT
023660  005201                              INC     R1              ;INCREMENT THE COUNT
023662  102001                              BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED
023664  005301                              DEC     R1              ;RESET THE COUNT TO 177777
023666  010137  002340          2$:         MOV     R1,TXINTC       ;SAVE NEW COUNT VALUE
023672                          60$:        PASS                    ;RESTORE GPRS.
023672  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023674  000002                              RTI
```

INTERRUPT SERVICE ROUTINE - CLKINT -

```
                                .SBTTL  INTERRUPT SERVICE ROUTINE       - CLKINT -
                                ;++ ************************************************************************
                                ;*      This routine is executed CLKHRZ times per second.  It decrements the
                                ;*      two timer counters down to zero.
                                ;*
                                ;* INPUTS:      TIMER1 - Timer counter #1.
                                ;*              TIMER2 - Timer counter #2.
                                ;*              TIMER3 - Timer counter for call of BREAK macro.
                                ;*
                                ;* OUTPUTS:     The 2 timer counters are decremented if they are not zero.
                                ;*
                                ;* CALLING SEQUENCE:    Put #CLKINT in the clock interrupt vector slot.
                                ;*                      Put the desired time period (seconds times CLKHRZ) in
                                ;*                      either TIMER1 or TIMER2 and poll the respective timer
                                ;*                      counter to detect its going to 0 on time-out.
                                ;*
                                ;* COMMENTS:    The 2 counters will not wraparound but will stop at 0.  This
                                ;*              allows the detection of a time-out any time after the time-out
                                ;*              has occurred until the timer counter is set to another value.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;-- ************************************************************************

023676  005737  002364        CLKINT:: TST     TIMER1          ;CHECK FOR TIMER1 AT ZERO.
023702  001402                         BEQ     2$              ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
023704  005337  002364                 DEC     TIMER1          ;DECREMENT TIME COUNT.
023710  005737  002366        2$:      TST     TIMER2          ;CHECK FOR TIMER2 AT ZERO.
023714  001402                         BEQ     4$              ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
023716  005337  002366                 DEC     TIMER2          ;DECREMENT TIME COUNT.
023722  005337  002370        4$:      DEC     TIMER3          ;DECREMENT THE BREAK COUNT.
023726  001006                         BNE     60$             ;EXIT IF NOT TIME TO CALL BREAK.
023730  013737  002372  002370        MOV     BCOUNT,TIMER3   ;SET UP TIME TILL NEXT BREAK.
023736  010046                         MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 FROM BREAK MACRO.
023740                                 BREAK                   ;CHECK FOR OPERATOR CONTROL/C.
023740  104422                                                         TRAP    C$BRK
023742  012600                         MOV     (SP)+,R0        ;RESTORE CONTENTS OF R0.
023744  000002                 60$:    RTI
```

```
                                .SBTTL  INTERUPT SERVICE ROUTINE        - RXBRRT -
                                ;++ ******************************************************************************
                                ;*      - BR Level Test Receive Interrupt Service Routine -
                                ;*      This service routine handles Receive Interrupts during the Interrupt
                                ;*      BR Level Test.  This routine counts the interrupt and sets a flag
                                ;*      to indicate that the interrupt has occurred.  It also checks the
                                ;*      flag which indicates that a TX interrupt has occurred.  If the TX
                                ;*      interrupt flag is set, this routine sets an interrupt order error
                                ;*      flag indicating that a transmit interrupt was serviced before a
                                ;*      simultaneous receive interrupt.
                                ;*
                                ;* INPUTS:      RXINTC - Holds the count of the number of RX interupts.
                                ;*              RXINTF - RX Interrupt flags.
                                ;*
                                ;* OUTPUTS:     RXINTC - Contains the updated interupt count.
                                ;*              RXINTF - RX Int flags:
                                ;*                       (Bit 0 set, bit 14 set if TXINTF bit 0 is set.)
                                ;*
                                ;* CALLING SEQUENCE:    Put the address of the label RXBRRT in the vector
                                ;*                      location.
                                ;*
                                ;* COMMENTS:    NOTE: The FIFO is purged by this routine.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;-- ******************************************************************************

023746                          RXBRRT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
023746  004537  003776                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023752  017700  156270                 MOV     @RBUFA,R0       ;READ THE CHAR OUT OF THE FIFO.
023756  013701  002334                 MOV     RXINTC,R1       ;GET THE INTERUPT COUNT.
023762  005201                         INC     R1              ;INCREMENT THE COUNT.
023764  001402                         BEQ     2$              ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
023766  010137  002334                 MOV     R1,RXINTC       ;SAVE NEW COUNT VALUE.
023772  013701  002336         2$:     MOV     RXINTF,R1       ;GET THE RX INTERRUPT FLAGS.
023776  052701  000001                 BIS     #BIT0,R1        ;SET THE RX INTERRUPT HAS OCCURRED FLAG.
024002  032737  000001  002342         BIT     #BIT0,TXINTF    ;TEST THE "TX INT HAS OCCURRED" FLAG.
024010  001402                         BEQ     4$              ;SKIP SETTING ERROR FLAG IF NO TX INT.
024012  052701  040000                 BIS     #BIT14,R1       ;SET THE INTERRUPT ORDER ERROR FLAG.
                                ;+
                                ; 8 FIFO codes will cause 8 interrupts, after these 8 codes we don't want
                                ;  to check the interrupt order, because perhaps a BMP code has come in
                                ;  between the servicing of the 8 FIFO code interrupts and the servicing
                                ;  of one of the TX interrupts.
                                ;-
024016  023737  002334  000010  4$:     CMP     RXINTC,NUMLNS   ;TEST FOR ALL SELFTEST CODE INTS DONE.
024024  003002                         BGT     60$             ;SKIP UPDATING RX INT FLAGS IF EXTRA RX INTS.
024026  010137  002336                 MOV     R1,RXINTF       ;UPDATE THE RX INTERRUPT FLAGS.
024032                          60$:    PASS                    ;RESTORE GPRS.
024032  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
024034  000002                         RTI
```

```
                                .SBTTL  INTERUPT SERVICE ROUTINE        - RXINPT -
                                ;++ ******************************************************************************
                                ;*      - Receive Character Input Interrupt Service Routine -
                                ;*      This service routine inputs a character from the DUT and loads the
                                ;*      char (complete with status flags) into a receive char buffer in
                                ;*      memory.  The interrupt is also counted.  The receive char buffer is
                                ;*      monitored to ensure that it does not overflow.
                                ;*
                                ;* INPUTS:      BUFEND - Labels the end of the host memory buffer.
                                ;*              BUFPTR - Contains address of next free buffer location.
                                ;*              CSRA - Contains the address of the DUT CSR.
                                ;*              RBUFA - Contains the address of the RBUF DUT register.
                                ;*              RXINTC - Holds the count of the number of RX interupts.
                                ;*              RXINTF - RX Interrupt flags.
                                ;*
                                ;* OUTPUTS:     BUFPTR - Contains updated address of next free buffer location.
                                ;*              RXINTC - Contains the updated interupt count.
                                ;*              RXINTF - RX Int flags (bit 15 set if RX.DATA.AVAIL is clear).
                                ;*
                                ;* CALLING SEQUENCE:    Put the address of the label RXINPT in the vector
                                ;*                      location.
                                ;*
                                ;* COMMENTS:    In case of overflow of the memory buffer, BUFPTR will be
                                ;*              maintained equal to BUFEND and the word at BURFPTR will be
                                ;*              the last word read from the DUT FIFO.
                                ;*              NOTE: This routine can destroy TX.ACTIONS by reading the CSR.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;-- ******************************************************************************

                                RXINPT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024036  004537  003776                     JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
     024042  032777  000200  156174             BIT     #BIT7,@CSRA     ;TEST RX.DATA.AVAIL. BIT OF THE CSR (READS CSR).
     024050  001003                             BNE     2$              ;BRANCH AROUND SETTING FLAG IF BIT IS SET.
     024052  052737  100000  002336             BIS     #BIT15,RXINTF   ;SET THE RX.DATA.AVAIL CLEAR FLAG.
     024060  013701  002334             2$:     MOV     RXINTC,R1       ;GET THE INTERUPT COUNT.
     024064  005201                             INC     R1              ;INCREMENT THE COUNT.
     024066  001402                             BEQ     4$              ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
     024070  010137  002334                     MOV     R1,RXINTC       ;SAVE NEW COUNT VALUE.
     024074  013702  002324             4$:     MOV     BUFPTR,R2       ;GET THE POINTER TO NEXT FREE BUFFER WORD.
     024100  017722  156142                     MOV     @RBUFA,(R2)+    ;READ A CHAR FROM THE FIFO INTO BUFFER.
     024104  020237  003726                     CMP     R2,BUFEND       ;TEST FOR POINTER BEYOND END OF BUFFER.
     024110  103002                             BHIS    60$             ;SKIP THE PTR UPDATE IF PTR OUT OF BOUNDS.
     024112  010237  002324                     MOV     R2,BUFPTR       ;UPDATE THE BUFFER POINTER.
     024116                             60$:    PASS                    ;RESTORE GPRS.
     024116  004736                             JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
     024120  000002                             RTI
```

GLOBAL TRAP SERVICE ROUTINE   - TP4RTN -

```
                                .SBTTL  GLOBAL TRAP SERVICE ROUTINE     - TP4RTN -
                                ;*************************************************************************
                                ;*      Bus Time-out Trap (004 trap) Service Routine -
                                ;*      This routine is used during the Device Register Address Access Test.
                                ;*      It determines if the 004 trap was caused by an "expected" error or
                                ;*      not by examining the return PC value on the stack.  If the trap is
                                ;*      unexpected, this routine jumps to the normal Diagnostic Supervisor
                                ;*      004 trap handling routine.
                                ;*
                                ;* INPUTS:      SP - Points to the PC where the trap occured.
                                ;*              ADRPTR - Label at the address where "expected" traps occur.
                                ;*              TP4FLG - 004 trap flags.
                                ;*
                                ;* OUTPUTS:     TP4FLG - Bit 15 is set if "expected" trap occured.
                                ;*
                                ;* CALLING SEQUENCE:    Put address pointed to by TP4RTN in 004 vector.
                                ;*                      Occurence of 004 trap vectors to this routine.
                                ;*
                                ;* COMMENTS:    Any 004 trap which occurs at an address other than that labeled
                                ;*              ADRPTR will be handled by the normal 004 trap service routine.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;*************************************************************************

024122  021627  020140          TP4RTN:: CMP     (SP),@ADRPTR    ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
024126  001402                           BEQ     2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
024130  000177  156210                   JMP     @TP4VEC         ;IF NOT,JUMP TO NORMAL 004 TRAP SERVICE RTN.
024134  052737  100000  002346  2$:      BIS     #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
024142  000002                           RTI                     ;ALL DONE, GO BACK TO THE TEST.
```

```
                        .SBTTL  INTERUPT SERVICE ROUTINE        - TXINTR
                        ;** ****************************************************************************
                        ;*      - Transmit Interrupt Service Routine -
                        ;*      This routine handles a transmit interrupt from the Device Under Test
                        ;*      (DUT) by counting the interrupt and reading the DUT CSR to clear the
                        ;*      interrupt request.  This routine also sets a flag to indicate that
                        ;*      a TX interrupt has occurred and sets a flag if the TX.ACTION bit is
                        ;*      not set in the read contents of the DUT CSR.
                        ;*
                        ;* INPUTS:      CSRA - Contains the address of the CSR.
                        ;*              TXINTC - Holds the count of the number of TX interupts.
                        ;*              TXINTF   TX Interrupt flags.
                        ;*
                        ;* OUTPUTS:     TXINTC - Contains the updated TX interupt count.
                        ;*              TXINTF   TX Int flags (bit 0 set, bit 15 set if TX.ACTION clr).
                        ;*
                        ;* CALLING SEQUENCE:    Put the address of the label TXINTR in the vector
                        ;*                      location.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: none
                        ;   ****************************************************************************

4508 024144             TXINTR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024144  004537  003776                     JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
4509 024150  013701  002340             MOV     TXINTC,R1       ;GET THE TX INTERUPT COUNT.
4510 024154  005201                     INC     R1              ;INCREMENT THE COUNT.
4511 024156  102001                    BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED.
4512 024160  005301                     DEC     R1              ;RESET THE COUNT TO 177777.
4513 024162  010137  002340     2$:     MOV     R1,TXINTC       ;SAVE NEW COUNT VALUE.
4514 024166  013703  002342             MOV     TXINTF,R3       ;GET THE TX INTERRUPT FLAGS.
4515 024172  017702  156046             MOV     @CSRA,R2        ;READ THE CSR.
4516 024176  100402                    BMI     4$              ;SKIP SETTING OF FLAG IF TX.ACTION IS SET.
4517 024200  052703  100000             BIS     #BIT15,R3       ;SET THE TX.ACTION CLEAR FLAG.
4518 024204  052703  000001     4$:     BIS     #BIT0,R3        ;SET THE TX INT HAS OCCURRED FLAG.
4519 024210  010337  002342             MOV     R3,TXINTF       ;UPDATE THE TX INTERRUPT FLAGS.
4520 024214             60$:    PASS                            ;RESTORE GPRS.
     024214  004736                             JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
4521 024216  000002                    RTI
```

```
                        ;**********************************************************************
                        ;
                        ;               VDHA.RPT
                        ;
                        ;**********************************************************************


                        .SBTTL  REPORT CODING SECTION

                        ;++
                        ; THE REPORT CODING SECTION CONTAINS THE
                        ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                        ;

024220                          BGNRPT
024220                                                          L$RPT::

024220                          EXIT    RPT
024220  000167                                                         .WORD   J$JMP
024222  000000                                                         .WORD   L10017-2-.

                                .EVEN

024224                          ENDRPT
024224                                                          L10017:
024224  104425                                                         TRAP    C$RPT
```

PROTECTION TABLE

```
                                .SBTTL  PROTECTION TABLE

                                ;*****************************************************************
*
                                ;
                                ;               FVTSKL4.P11
                                ;
                                ;*****************************************************************



                                ;++
                                ; THIS TABLE IS USED BY THE RUNTIME SERVICES
                                ; TO PROTECT THE LOAD MEDIA.
                                ;--

024226                                  BGN?ROT
024226                                                                  L$PROT::

024226  177777                          -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
024230  177777                          -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
024232  177777                          -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

024234                                  ENDPROT
```

```
                        ;*****************************************************************************
                        ;
                        ;               VDHA.INI
                        ;
                        ;*****************************************************************************


                        .SBTTL  INITIALIZE SECTION
                        ;++
                        ;*****************************************************************************
                        ;*      This section contains the code which is performed at the beginning of
                        ;*      each pass or after a continue command.
                        ;*      This code performs the following actions:
                        ;*
                        ;*      Moves the information held in the hardware P table into the global
                        ;*      data area.
                        ;*
                        ;*****************************************************************************
                        ;--
024234                          BGNINIT                                         L$INIT::
024234
                        ;SEE IF PROGRAM JUST STARTED, BR IF YES
024234                          READEF  @EF.START
024234  012700  000040                                                          MOV     @EF.START,R0
024240  104447                                                                  TRAP    C$REFG
024242                          BCOMPLETE       NEWSTA
024242  103416                                                                  BCS     NEWSTA
                        ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
024244                          READEF  @EF.RESTART
024244  012700  000037                                                          MOV     @EF.RESTART,R0
024250  104447                                                                  TRAP    C$REFG
024252                          BCOMPLETE       NEWRES
024252  103555                                                                  BCS     NEWRES
                        ;SEE IF THIS IS A NEW PASS, BR IF YES
024254                          READEF  @EF.NEW
024254  012700  000035                                                          MOV     @EF.NEW,R0
024260  104447                                                                  TRAP    C$REFG
024262                          BCOMPLETE       NEWPAS
024262  103554                                                                  BCS     NEWPAS
                        ;SEE IF PROGRAM WAS JUST CONTINUED
024264                          READEF  @EF.CONTINUE
024264  012700  000036                                                          MOV     @EF.CONTINUE,R0
024270  104447                                                                  TRAP    C$REFG
024272                          BNCOMPLETE      GETPRM
024272  103160                                                                  BCC     GETPRM
024274  000137  025050          JMP     ENDIT
024300                  NEWSTA:
024300                          BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
024300  104433                                                                  TRAP    C$RESET
                        ;++
                        ; Set up for Line Time Clock interrupts.
                        ;
024302                          CLOCK   L,R1            ;GET THE CLOCK PARAMETERS.
024302  012700  000114                                                          MOV     @ L,R0
024306  104462                                                                  TRAP    C$CLCK
```

INITIALIZE SECTION

```
        024310  010001                                                          MOV     R0,R1
 4625   024312  012137  002354                  MOV     (R1)+,CLKCSR            ;STORE CLOCK CSR ADDRESS.
 4626   024316  012137  002356                  MOV     (R1)+,CLKBRL            ;STORE CLOCK BUS REQ INT LEVEL.
 4627   024322  012137  002360                  MOV     (R1)+,CLKVEC            ;STORE CLOCK INTERRUPT VECTOR.
 4628   024326  012137  002362                  MOV     (R1)+,CLKHRZ            ;STORE CLOCK FREQUENCY.
 4629   024332  023727  002362  000062          CMP     CLKHRZ,#50.             ;TEST FOR 50HZ LINE FREQUENCY.
 4630   024340  001004                          BNE     2$                      ;BRANCH IF CLOCK IS NOT 50HZ.
 4631   024342  012737  000024  002374          MOV     #20.,MSTICK             ;INDICATE 20MS PER CLOCK TICK.
 4632   024350  000403                          BR      4$
 4633   024352  012737  000021  002374  2$:     MOV     #17.,MSTICK             ;INDICATE 17 MS PER CLOCK TICK.
 4634   024360                          4$:     SETVEC  CLKVEC,#CLKINT,#PRI06   ;INITIALIZE CLOCK INTERRUPT VECTOR.
        024360  012746  000300                                                  MOV     #PRI06,-(SP)
        024364  012746  023676                                                  MOV     #CLKINT,-(SP)
        024370  013746  002360                                                  MOV     CLKVEC,-(SP)
        024374  012746  000003                                                  MOV     #3,-(SP)
        024400  104437                                                          TRAP    C$SVEC
        024402  062706  000010                                                  ADD     #10,SP
 4635   024406  013700  002362                  MOV     CLKHRZ,R0               ;INITIALIZE THE BREAK COUNT
 4636   024412  006200                          ASR     R0                      ; TO CAUSE A BREAK
 4637   024414  010037  002372                  MOV     R0,BCOUNT               ; EVERY 1/2 SECOND.
 4638   024420  106427  000240                  MTPS    #PRI05                  ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
 4639                                  ;+
 4640                                  ; Enable the Line Time Clock (LTC) checking to make sure that the CSR
 4641                                  ; is accessable.
 4642                                  ;    First set up to catch any 004 traps which occur:
 4643                                  ;-
 4644   024424  013737  000004  002344          MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
 4645   024432  012737  024122  000004          MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
 4646                                  ;+
 4647                                  ;    Enable LTC checking for 004 trap in case CSR is not there.
 4648                                  ;-
 4649   024440  005037  002346                  CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
 4650   024444  012737  000100  002350          MOV     #BIT6,WORD1     ;SET UP TO SET BIT6 OF THE LTC CSR.
 4651   024452  012700  002350                  MOV     #WORD1,R0       ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
 4652   024456  013701  002354                  MOV     CLKCSR,R1       ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
 4653   024462  004737  020126                  JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
 4654   024466  013737  002344  000004          MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
 4655   024474  103403                          BCS     6$              ;IF NO TRAP, LTC IS THERE SO CONTINUE.
 4656   024476  005037  002362                  CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
 4657   024502  000402                          BR      8$              ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
 4658                                  ;+
 4659                                  ; Calibrate the DELAY routine milli-second delay count value.
 4660                                  ;-
 4661   024504  004737  017702          6$:     JSR     PC,CALMSL
 4662                                  ;+
 4663                                  ; Check for Memmory Management present on this machine.
 4664                                  ; If MEM MGT is present, disable it.
 4665                                  ;-
 4666   024510  013737  000004  002344  8$:     MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
 4667   024516  012737  024122  000004          MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
 4668   024524  005037  002346                  CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
 4669   024530  005037  002350                  CLR     WORD1           ;PREPARE TO CLEAR THE MEM MGT SR0 REGISTER.
 4670   024534  012700  002350                  MOV     #WORD1,R0       ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
 4671   024540  013701  002400                  MOV     MMSR0,R1        ;SELECT MEM MGT SR0 REGISTER AS DESTINATION.
 4672   024544  005037  002402                  CLR     MMPRES          ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
 4673   024550  005037  002404                  CLR     MMENAB          ;INDICATE MEM MGT IS NOT ENABLED.
 4674   024554  004737  020126                  JSR     PC,CKTRAP       ;CLEAR THE MEM MCT SR0 REG AND CHECK FOR TRAP.
```

INITIALIZE SECTION SEQ 0120

```
 4675 024560  013737  002344  000004          MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
 4676 024566  103003                          BCC     10$             ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
 4677 024570  012737  000001  002402          MOV     #1,MMPRES       ;INDICATE THAT MEM MGT IS PRESENT.
 4678 024576  005037  002332          10$:    CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
 4679 024602  000137  024614                  JMP     NEWPAS          ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
 4680
 4681 024606                          NEWRES: BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
      024606  104433                                                                  TRAP    C$RESET
 4682 024610  005037  002332                  CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
 4683 024614                          NEWPAS:
 4684 024614  012737  177777  002240          MOV     #-1,UNITN       ;RESET LOGICAL DEVICE TO -1
 4685                                 ;+
 4686                                 ; Increment the pass counter, correct for any overflow.
 4687                                 ;  This counter is used in the Rom version test.
 4688                                 ;
 4689 024622  005237  002332                  INC     PASCNT          ;INCREMENT THE PASS COUNTER.
 4690 024626  001002                          BNE     GETPRM          ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
 4691 024630  005337  002332                  DEC     PASCNT          ;SET PASS COUNT TO 177777 OCTAL.
 4692
 4693                                 ; GET THE HAPDWARE PARAMETERS FOR THIS UNIT.
 4694 024634                          GETPRM:
 4695 024634  005237  002240                  INC     UNITN           ;INCREMENT LOGICAL DEVICE NUMBER
 4696 024640  023737  002240  002012          CMP     UNITN,L$UNIT    ;SEE IF MAXIMUM UNIT NO. EXCEEDED
 4697 024646  002362                          BGE     NEWPAS          ;BR IF YES
 4698
 4699 024650                                  GPHARD  UNITN,R1        ;GET P-TABLE POINTER INTO R1
      024650  013700  002240                                                          MOV     UNITN,R0
      024654  104442                                                                  TRAP    C$GPHRD
      024656  010001                                                                  MOV     R0,R1
 4700 024660                                  BCOMPLETE       30$     ;BR IF DEVICE AVAILABLE
      024660  103401                                                                  BCS     30$
 4701 024662  000764                          BR      GETPRM          ;SKIP THIS DEVICE
 4702
 4703
 4704                                 ;************ HARDWARE PARAMETER MOVING CODE ****************
 4705 024664  012137  002244          30$:    MOV     (R1)+,CSRA      ;STORE DHV11 M CSR ADDRESS IN DEV.REG.ADDRESS TABLE
 4706 024670  012102                          MOV     (R1)+,R2        ;GET THE RX INTERRUPT VECTOR ADDRESS.
 4707 024672  010237  002232                  MOV     R2,RXVECA       ;STORE RX INT VECTOR ADDRESS.
 4708 024676  062702  000004                  ADD     #4,R2           ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
 4709 024702  010237  002234                  MOV     R2,TXVECA       ;STORE TX INT VECTOR ADDRESS.
 4710 024706  012137  002236                  MOV     (R1)+,ACTLNS    ;STORE DHV11-M ACTIVE LINE BIT MAP
 4711 024712  012702  000377                  MOV     #MAPLNS,R2      ;GET THE BIT MAP FOR ALL LINES.
 4712 024716  005102                          COM     R2              ;GET A BIT MAP OF NON-EXISTANT LINES.
 4713 024720  040237  002236                  BIC     R2,ACTLNS       ;CLEAR NON-EXISTANT LINES FROM ACTLNS.
 4714 024724  112137  002242                  MOVB    (R1)+,BRLEVL    ;STORE DHV11-M INTERUPT BUS REQUEST LEVEL
 4715                                 ;+
 4716                                 ;       CALCULTATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
 4717                                 ;       DEVICE REGISTER ADDRESS TABLE.
 4718                                 ;
 4719 024730  013701  002244                  MOV     CSRA,R1         ;COPY CSR ADDRESS
 4720 024734  005201                          INC     R1              ;INCREMENT CSR ADDRESS
 4721 024736  005201                          INC     R1              ; COPY BY 2.
 4722 024740  012703  000007                  MOV     #7,R3           ;SET UP REGISTER COUNT
 4723 024744  012702  002246                  MOV     #RBUFA,R2       ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
 4724 024750  010122                  12$:    MOV     R1,(R2)+        ;STORE REGISTER ADDRESS IN TABLE
 4725 024752  005201                          INC     R1              ;INCREMENT REGISTER ADDRESS
 4726 024754  005201                          INC     R1              ; BY 2,FOR THE NEXT DEVICE REGISTER.
```

INITIALIZE SECTION

```
 4727 024756  005303                        DEC     R3              ;DECREMENT REGISTER COUNT
 4728 024760  001373                        BNE     12$             ;LOOP IF NOT DONE
 4729
 4730                                ;+
 4731                                ; Initialise the BMP code queue.
 4732                                ;-
 4733 024762  012700  002526                 MOV     #BMPCQB,R0      ;GET THE START ADDRESS OF THE QUEUE.
 4734 024766  012701  002726                 MOV     #BMPCQE,R1      ;GET THE END ADDRESS OF THE QUEUE.
 4735 024772  010037  002524                 MOV     R0,BMPCQP       ;SET THE POINTER TO THE START OF THE QUEUE.
 4736 024776  005020                 14$:    CLR     (R0)+           ;CLEAR OUT THE CONTENTS OF THE QUEUE.
 4737 025000  020001                         CMP     R0,R1           ;CHECK IF END OF QUEUE HAS BEEN REACHED.
 4738 025002  103775                        BLO     14$             ;LOOP IF NOT ALL DONE.
 4739                                ;+
 4740                                ; Report the Unit number if the software P table question was answered YES,
 4741                                ;  and the maximum unit number is greater than 1.
 4742                                ;
 4743 025004  032737  000020  002226         BIT     #BIT4,OPTION    ;CHECK IF THE QUESTION WAS ANSWERED YES.
 4744 025012  001416                        BEQ     16$             ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
 4745 025014  023727  002012  000001         CMP     L$UNIT,#1       ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
 4746 025022  003412                        BLE     16$             ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
 4747 025024                                PRINTF  #MFUNIT,UNITN   ;REPORT UNIT NUMBER.
      025024  013746  002240                                                    MOV     UNITN,-(SP)
      025030  012746  004076                                                    MOV     #MFUNIT,-(SP)
      025034  012746  000002                                                    MOV     #2,-(SP)
      025040  010600                                                            MOV     SP,R0
      025042  104417                                                            TRAP    C$PNTF
      025044  062706  000006                                                    ADD     #6,SP
 4748 025050                         16$:
 4749
 4750 025050  005037  002352         ENDIT:  CLR     CTRLCF          ;CLR THE CTRL-C TEST ABORT FLAG.
 4751                                ;+
 4752                                ; Set the processor priority to allow LTC interrupts but not others.
 4753                                ;-
 4754 025054  106427  000240                 MTPS    #PRI05                  ;SET PROCESSOR PRIORITY TO 5.
 4755                                ;+
 4756                                ;    Enable L ne Time Clock if one is available.
 4757                                ;-
 4758 025060  005737  002362                  TST    CLKHRZ          ;CHECK FOR A LTC BEING PRESENT.
 4759 025064  001403                        BEQ     18$             ;LTC PRESENT?  NO, SKIP LTC ENABLE.
 4760 025066  012777  000100  155260          MOV    #BIT6,@CLKCSR   ;YES, ENABLE THE LTC.
 4761 025074                         18$:
 4762
 4763 025074                                ENDINIT
      025074                                                            L10021:
      025074  104411                                                           TRAP    C$INIT
 4764
 4765         000000                        TNUM == 0               ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.
```

```
 4768                           ;*****************************************************************************
 4769                           ;
 4770                           ;               VDHA.ATD
 4771                           ;
 4772                           ;*****************************************************************************
 4774
 4775
 4776                           .SBTTL  AUTODROP SECTION
 4777
 4778
 4779                           ;++
 4780                           ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
 4781                           ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
 4782                           ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
 4783                           ; DROPPED FROM TESTING.
 4784                           ;--
 4785
 4786 025076                           BGNAUTO
      025076                                                          L$AUTO::
 4787
 4794
 4795 025076                           ENDAUTO
      025076                                                          L10022:
      025076  104461                                                         TRAP    C$AUTO
```

AUTODROP SECTION

```
4797
4798                            ;*****************************************************************************************
4799                            ;
4800                            ;               VDHA.CUC
4801                            ;
4802                            ;*****************************************************************************************
4803
4804
4805
4806                            .SBTTL  CLEANUP CODING SECTION
4807
4808                            ;++
4809                            ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4810                            ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
4811                            ;--
4812
4813 025100                             BGNCLN
     025100                                                                 L$CLEAN::
4814
4823 025100  005737  002352             TST     CTRLCF          ;DID WE GET HERE BY CTRL-C FROM TEST?
4824 025104  001403                     BEQ     2$              ;CTRL-C FROM TEST?  NO, SKIP BUS RESET.
4825 025106                             BRESET                  ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
     025106  104433                                                    TRAP    C$RESET
4826 025110  106427  000240             MTPS    @PRIO5          ;DISABLE DEVICE INTERRUPTS, ALLOW LTC INTS.
4827 025114                     2$:
4828 025114  005737  002362             TST     CLKHRZ
4829 025120  001402                     BEQ     3$
4830 025122  005077  155226             CLR     @CLKCSR
4831 025126                     3$:
4832 025126                             EXIT    CLN
     025126  104432                                                    TRAP    C$EXIT
     025130  000002                                                    .WORD   L10023-.
4833
4845
4846                                    .EVEN
4847
4848 025132                             ENDCLN
     025132                                                         L10023:
     025132  104412                                                    TRAP    C$CLEAN
```

CLEANUP CODING SECTION

```
                        ;*****************************************************************************
                        ;
                        ;               \DHA.DRF
                        ;
                        ;*****************************************************************************



                        .SBTTL  DROP UNIT SECTION

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

                                SGNDU
                                                                        L$DU::

                        ;*****************************************************************************
                        ;       INSERT DROP CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
                        ;       A "DROP" COMMAND OR A "DODU" MACRO EXECUTION.  THE PURPOSE
                        ;       OF THIS CODE IS TO DO ANY NECESSARY HOUSEKEEPING AFTER A
                        ;       UNIT HAS BEEN DROPPED.  THIS SECTION IS OPTIONAL.
                        ;*****************************************************************************

                                 PRINTF #DROP,R0        ;REPORT UNIT THAT HAS BEEN DROPPED.
025134  010046                                                         MOV     R0,-(SP)
025136  012746  025160                                                 MOV     #DROP,-(SP)
025142  012746  000002                                                 MOV     #2, (SP)
025146  010600                                                         MOV     SP,R0
025150  104417                                                         TRAP    C$PNTF
025152  062706  000006                                                 ADD     #6,SP
025156  000427                  BR      EDROP           ;BRANCH AROUND THE MESSAGE.

025160     045     101     040  DROP:   .ASCIZ/%A UNIT%D6%A DROPPED FROM FURTHER TESTING.%N/
025163     125     116     111
025166     124     045     104
025171     066     045     101
025174     040     104     122
025177     117     120     120
025202     105     104     040
025205     106     122     117
025210     115     040     106
025213     125     122     124
025216     110     105     122
025221     040     124     105
025224     123     124     111
025227     116     107     056
025232     045     116     000

                        .EVEN
025236                  EDROP:
025236                  EXIT    DU
025236  000167                                                         .WORD   J$JMP
025240  000000                                                         .WORD   L10024-2-.
```

DROP UNIT SECTION

SEQ 0125

```
   4884
   4885 025242                              ENDDU
        025242                                                    L10024:
        025242  104453                                                    TRAP    C$DU
```

DROP UNIT SECTION

```
4887
4888                            ;*****************************************************************************************
4889                            ;
4890                            ;               VDHA.ADD
4891                            ;
4892                            ;*****************************************************************************************
4893
4894
4895
4896                            .SBTTL  ADD UNIT SECTION
4897
4898                            ;++
4899                            ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
4900                            ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
4901                            ;  TO THE TEST CYCLE.
4902                            ;--
4903
4904 025244                             BGNAU
    025244                                                                  L$AU::
4905
4906                            ;*****************************************************************************************
4907                            ;       INSERT ADD CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
4908                            ;       AN "ADD" COMMAND.  THE PURPOSE OF THIS CODE IS TO DO ANY
4909                            ;       HOUSEKEEPING THAT MAY BE NECESSARY AFTER A UNIT HAS BEEN ADDED.
4910                            ;       THIS SECTION IS OPTIONAL.
4911                            ;*****************************************************************************************
4912
4913 025244                             EXIT    AU
    025244  000167                                                          .WORD   J$JMP
    025246  000000                                                          .WORD   L10025-2-.
4914
4915
4916                                    .EVEN
4917
4918 025250                             ENDAU
    025250                                                              L10025:
    025250  104452                                                          TRAP    C$AU
4919
```

```
                                .SBTTL  HARDWARE TEST           - ADRA -
                                ;++
                                ;*****************************************************************************
                                ;*                       - REGISTER ADDRESS TEST -
                                ;*
                                ;*      This test verifies that the Q-bus can read and write to the DHV11
                                ;*      device registers.  If the DHV11 does not respond to the access
                                ;*      attempts (If the DHV11 is at the wrong address, for example) the
                                ;*      004 bus time-out trap is detected by this routine and an error
                                ;*      is reported.
                                ;*
                                ;*****************************************************************************
                                ;--

025252                                  BGNTST
025252                                                                  T1::
        000001                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025252  012737  000001  002326          MOV     @TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (1)
025260  012737  177777  002352          MOV     @-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
                                ;+
                                ; Set up to catch any 004 traps which occur:
                                ;-
025266  013737  000004  002344          MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
025274  012737  024122  000004          MOV     @TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
025302  005005                          CLR     R5              ;CLEAR THE ERROR FLAGS.

                                ;+
                                ; Set up for the initial iteration of the test loop:
                                ;-
025304  005004                          CLR     R4              ;CLEAR THE LINE COUNTER.

                                ;+
                                ; Here begins the loop to test the registers for a line.
                                ;    First test the CSR and set the IND.ADR.REG (I.A.R) field.
                                ;-
025306  005037  002346          2$:     CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
025312  013700  002244                  MOV     CSRA,R0         ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
025316  012701  025532                  MOV     #52$,R1         ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
025322  004737  020126                  JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
025326  103402                          BCS     4$              ;IF NO TRAP, BYPASS ERROR.
025330  052705  100001                  BIS     #100001,R5      ;SET FATAL READ ERROR FLAGS.
025334  042737  000017  025532  4$:     BIC     #17,52$         ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
025342  050437  025532                  BIS     R4,52$          ;OR IN THE LINE COUNTER TO THE I.A.R FIELD.
025346  010100                          MOV     R1,R0           ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
025350  013701  002244                  MOV     CSRA,R1         ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
025354  004737  020126                  JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
025360  103403                          BCS     6$              ;IF NO TRAP, BYPASS ERROR.
025362  052705  100002                  BIS     #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
025366  000440                          BR      40$             ;EXIT AND REPORT FATAL ERROR.
                                ;+
                                ;    Now, we test each register for this line.
                                ;-
025370  012702  000010          6$:     MOV     #10,R2          ;INIT REGISTER COUNTER TO 8.
025374  013737  002244  025530          MOV     CSRA,50$                ;INITIALIZE THE REGISTER POINTER.
025402  012700  025530          8$:     MOV     #50$,R0         ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
025406  012701  025532                  MOV     #52$,R1         ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
025412  004737  020126                  JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
```

```
 4977 025416  103402                        BCS     10$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
 4978 025420  052705  100001                 BIS     #100001,R5      ;SET FATAL READ ERROR FLAGS.
 4979 025424  010100                 10$:    MOV     R1,R0           ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
 4980 025426  012701  025530                 MOV     #50$,R1         ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.
 4981 025432  004737  020126                 JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
 4982 025436  103402                        BCS     12$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
 4983 025440  052705  100002                 BIS     #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
 4984 025444  005237  025530         12$:    INC     50$             ;INCREMENT THE REGISTER
 4985 025450  005237  025530                 INC     50$             ; POINTER BY 2.
 4986 025454  005302                        DEC     R2              ;COUNT THE REGISTER.
 4987 025456  001351                        BNE     8$              ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
 4988
 4989                                ;+
 4990                                ; Now we set up to test the next line, or to exit if we are done.
 4991                                ;
 4992 025460  005204                        INC     R4              ;INCREMENT THE LINE COUNTER.
 4993 025462  020427  000010                CMP     R4,#NUMLNS      ;COMPARE LINE COUNTER AGAINST NUMBER OF LINES.
 4994 025466  002707                        BLT     2$              ;LOOP TO TEST THE NEXT LINE IF WE'RE NOT DONE.
 4995
 4996                                ;+
 4997                                ; Done cnecking device register addresses.
 4998                                ;    Report any errors and ex't.
 4999                                ;-
 5000 025470  013737  002344  000004 40$:    MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
 5001 025476  005705                        TST     R5              ;CHECK THE ERROR FLAGS.
 5002 025500  100015                        BPL     60$             ;EXIT ROUTINE IF NO ERRORS.
 5003                                    ; REPORT "DEVICE REGISTER ACCESS ERRORS"
 5004 025502                                ERRDF   101,EM0103,ER0101;              >>>>> ERROR #101 <<<<<.
      025502  104455                                                                    TRAP    C$ERDF
      025504  000145                                                                    .WORD   101
      025506  006067                                                                    .WORD   EM0103
      025510  016070                                                                    .WORD   ER0101
 5005
 5006 025512                                DODU    UNITN           ;DROP THIS UNIT FROM FUTHER TESTING.
      025512  013700  002240                                                            MOV     UNITN,R0
      025516  104451                                                                    TRAP    C$DODU
 5007 025520  005037  002352                 CLR     CTRLCF          ;INDICATE NO CTRL-C ABORT FROM TEST.
 5008 025524                                DOCLN                   ;ABORT THIS SUB PASS.
      025524  104444                                                                    TRAP    C$DCLN
 5009 025526  000402                        BR      60$             ;
 5010                                ;+
 5011                                ; Local storage.
 5012                                ;-
 5013 025530  000000                 50$:    .WORD   0       ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 5014 025532  000000                 52$:    .WORD   0       ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 5015 025534  005037  002352         60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 5016 025540                                ENDTST
      025540                                                            L10026:
      025540  104401                                                                   TRAP    C$ETST
```

```
                        .SBTTL  HARDWARE TEST           - MRSTA -
                        ;++ ******************************************************************************
                        ;*              - Master Reset With Selftest Test -
                        ;*      This test verifies that the Master Reset bit will clear after a device
                        ;*      reset and the performance of the DUT ROM based selftest.
                        ;*
                        ;-- ******************************************************************************
                                BGNTST
                                                                                        T2::
                                TNUM == TNUM + 1        ;INCREMENT THE A SEMBLY TIME TEST COUNTER.
                                 MOV    @TNUM,TSTNUM    ;SET UP THE TEST  UMBER.        (2)
                                 MOV    @-1,CTRLCF      ;INDICATE THAT h  ARE IN A TEST.
                                 MOV    @1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
                                 MOV    @EM0201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                 MOV    @ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                        ;+
                        ; Wait up to 5 seconds for the DUT Master Reset bit to clear.
                        ;-
                                 MOV    @5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
                                 MOV    @BIT05,R2       ;WAITING FOR MASTER RESET BIT.
                                 CLR    R3              ;WAITING FOR BIT TO CLEAR.
                                 MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
                                JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
                                 BCS    2$              ;SKIP TO RESET DUT IF MR CLEAR.
                        ;+
                        ; DUT Master Reset bit did not go clear.  Device may be stuck in some
                        ;  odd state.  Try to reset device with a bus reset.
                        ;-
                                 BRESET                 ;NO, TRY TO JOG DEVICE WITH BUS RESET.
                                                                                  TRAP    C$RESET
                                JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
                                 MOV    @5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
                                JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
                                 BCC    4$              ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
                        ;+
                        ; Set the Master Reset bit and verify that it clears within the proper time.
                        ;-
                        2$:      MOV    @5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
                                 MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
                                JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
                                 BCC    4$              ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
                                 MOV    @5000.,R2
                                 SUB    R1,R2           ;CALCULATE # OF MS FOR MR TO CLEAR.
                                 BEQ    6$              ;GO REPORT ERROR IF MR CLEAR IMMEDIATELY.
                                 CMP    R2,@500.
                                BLT     8$              ;GO REPORT ERROR IF MR CLEAR IN < 1/2 SECOND.
                                 BR     60$             ;EXIT THE TEST WITHOUT ERROR.
                        ;+
                        ; Error reports:
                        ;-
                            ;Report MR bit would not clear after a DUT reset.
                        4$:      MOV    @201.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
                                 MOV    @EM0202,R1      ;SELECT ERROR MESSAGE.
                                ERROR                   ;REPORT ERROR.          >>>>> ERROR @201 <<<<<
                                                                                  TRAP    C$ERROR
                                BR      60$             ;EXIT THE TEST.
```

HARDWARE TEST - MRSTA - SEQ 0130

```
 5072                                          ;Report MR bit clear immediately after DUT reset.
 5073 025716  012737  000312  003770  6$:      MOV     #202.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
 5074 025724  012701  006346                   MOV     #EM0203,R1      ;SELECT ERROR MESSAGE.
 5075 025730                                   ERROR                   ;REPORT ERROR.          >>>>> ERROR #202 <<<<<
      025730  104460                                                                                 TRAP    C$ERROR
 5076 025732  000406                           BR      60$             ;EXIT THE TEST.
 5077
 5078                                          ;Report MR clear within 1/2 second of DUT reset.
 5079 025734  012737  000313  003770  8$:      MOV     #203.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
 5080 025742  012701  006511                   MOV     #EM0204,R1      ;SELECT ERROR MESSAGE.
 5081 025746                                   ERROR                   ;REPORT ERROR.          >>>>> ERROR #203 <<<<<
      025746  104460                                                                                 TRAP    C$ERROR
 5082
 5083 025750  005037  002352          60$:     CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5084
 5085 025754                                   ENDTST
      025754                                                                           L10027:
      025754  104401                                                                                 TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - MRSSTA -
                                ;** ****************************************************************************
                                ;*              - Master Reset With Skip Selftest Test -
                                ;*      This test verifies that the Master Reset bit will clear after a device
                                ;*      reset and the skipping of the DUT ROM based selftest.
                                ;*
                                ;-- ****************************************************************************
                                        BGNTST
                                                                                        T3::
                                        TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 5096 025756  012737  000003  002326     MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (3)
 5097 025764  012737  177777  002352     MOV     # 1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
 5098 025772  012737  000001  003766     MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 5099 026000  012737  006670  003772     MOV     #EM0301,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
 5100 026006  012737  016406  00?  '4     MOV     #ER0201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
 5101                                   ;*
                                ; Wait up to 5 seconds for the DUT Master Reset bit to clear.
                                ;-
 5104 026014  012701  011610             MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
 5105 026020  012702  000040             MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
 5106 026024  005003                     CLR     R3             ;WAITING FOR BIT TO CLEAR.
 5107 026026  013704  002244             MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
 5108 026032  004737  020340            JSR     PC,MSLGET      ;WAIT FOR DUT CSR_MR BIT TO CLEAR.
 5109 026036  103410                     BCS     2$             ;SKIP TO RESET DUT IF MR CLEAR.
 5110                                   ;*
                                ; DUT Master Reset bit did not go clear.  Device may be stuck in some
                                ;  odd state.  Try to reset device with a bus reset.
                                ;-
 5114 026040                            BRESET                 ;NO, TRY TO JOG DEVICE WITH BUS RESET.
      026040  104433                                                    TRAP    C$RESET
 5115 026042  004737  022330            JSR     PC,SKPSTS      ;TRY TO SKIP THE SELFTEST.
 5116 026046  012701  011610             MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
 5117 026052  004737  020340            JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
 5118 026056  103024                     BCC     6$             ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
 5119                                   ;*
                                ; Set the Master Reset bit, try to skip the selftest, and verify that the
                                ;  MR bit clears within 1/5 second.
                                ;-
 5123 026060  012701  000310  2$:       MOV     #200.,R1       ;TIME-OUT VALUE IS 1/5 SECOND.
 5124 026064  010214                     MOV     R2,(R4)        ;SET THE DUT MASTER RESET BIT.
 5125 026066  004737  022330            JSR     PC,SKPSTS      ;TRY TO SKIP THE SELFTEST.
 5126 026072  004737  020340            JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
 5127 026076  103007                    BCC     4$             ;GO FIND OUT WHAT IS WRONG IF MR NOT CLEAR.
 5128 026100  012702  000310             MOV     #200.,R2
 5129 026104  160102                     SUB     R1,R2          ;CALCULATE # OF MS FOR MR TO CLEAR.
 5130 026106  020227  000012             CMP     R2,#10.
 5131 026112  002415                    BLT     8$             ;GO REPORT ERROR IF MR CLEAR IN < 10 MS.
 5132 026114  000431                     BR      60$            ;EXIT THE TEST WITHOUT ERROR.
 5133                                   ;*
                                ; MR did not clear within 1/5 second, see .f it clears within 5 seconds.
                                ;-
 5136 026116  012701  011300  4$:       MOV     #4800.,R1      ;TIME-OUT VALUE IS 5 SECONDS MINUS 1/5 SECOND.
 5137 026122  004737  020340            JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
 5138 026126  103416                     BCS     10$            ;GO REPORT ERROR IF MR CLEARED FINALLY.
 5139                                   ;*
                                ; Error reports:
                                ;-
```

HARDWARE TEST - MRSSTA -

```
5142                                        ;Report MR bit would not clear after a DUT reset.
5143 026130  012737  000455  003770  6$:     MOV     #0301.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5144 026136  012701  006173                 MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
5145 026142                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0301 <<<<<
     026142  104460                                                                              TRAP    C$ERROR
5146 026144  000415                         BR      60$             ;EXIT THE TEST.
5147
5148                                        ;Report MR bit clear within 10 ms after DUT reset.
5149 026146  012737  000456  003770  8$:     MOV     #0302.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5150 026154  012701  006733                 MOV     #EM0302,R1      ;SELECT ERROR MESSAGE.
5151 026160                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0302 <<<<<
     026160  104460                                                                              TRAP    C$ERROR
5152 026162  000406                         BR      60$             ;EXIT THE TEST.
5153
5154                                        ;Report MR cleared between 1/5 second and 5 seconds of DUT reset.
5155 026164  012737  000457  003770  10$:    MOV     #0303.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5156 026172  012701  007073                 MOV     #EM0303,R1      ;SELECT ERROR MESSAGE.
5157 026176                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0303 <<<<<
     026176  104460                                                                              TRAP    C$ERROR
5158
5159 026200  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5160
5161 026204                                 ENDTST
     026204                                                                     L10030:
     026204  104401                                                                              TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RXCHRA -
                                ;** ************************************************************************
                                ;*              - RBUF Register RX Character Field Test -
                                ;*      This test verifies that the RX Character Field of the DUT RBUF register
                                ;*      appears to be functioning correctly.  This test uses the codes which
                                ;*      should be in the FIFO after a board reset and skip selftest sequence.
                                ;*
                                ;   ************************************************************************
                                    BGNTST
                                                                                        T4::
                                     TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026206  012737  000004  002326        MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (4)
026214  012737  177777  002352        MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
026222  012737  000001  003766        MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026230  012737  007252  003772        MOV     #EM0401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026236  012737  016406  003774        MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Set the DUT CSR Master Reset (MR) bit, perform the skip selftest sequence,
                                ;  and wait up to 5 seconds for the MR bit to clear.
                                ;
026244  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
026250  012702  000040                  MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
026254  005003                          CLR     R3              ;WAITING FOR BIT TO CLEAR.
026256  013704  002244                  MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
026262  010214                          MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
026264  004737  022330                 JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
026270  004737  020340                 JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026274  103015                         BCC     4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; Read 6 characters from the DUT and verify that they are valid selftest
                                ;  codes.
                                ;-
026276  012400                          MOV     (R4)+,R0        ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
026300  012701  000006                  MOV     #6,R1           ;INITIALIZE THE LOOP COUNTER.
026304  011402                 2$:      MOV     (R4),R2         ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
026306  010200                          MOV     R2,R0
026310  042700  177476                  BIC     #177476,R0      ;REMOVE ALL BUT BITS SPECIFIC TO SELFTEST CODE.
026314  020027  000201                  CMP     R0,#201         ;CHECK THAT BITS 0,6, AND 7 ARE CORRECT.
026320  001012                         BNE     6$              ;GO REPORT ERROR IF CODE IS NOT SELFTEST CODE.
026322  005301                          DEC     R1              ;COUNT THIS LOOP ITERATION.
026324  001367                         BNE     2$              ;LOOP IF NOT ALL LINES DONE.
026326  000415                         BR      60$             ;EXIT TEST, NO ERROR FOUND.
                                ;+
                                ; Error .eports:
                                ;-
                                    ;Report MR bit would not clear after a DUT reset.
026330  012737  000621  003770  4$:       MOV     #0401.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026336  012701  006173                    MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
026342                                   ERROR                   ;REPORT ERROR.          >>>>> ERROR #0401 <<<<<
026342  104460                                                                              TRAP    C$ERROR
026344  000406                           BR      60$             ;EXIT THE TEST.

                                    ;Report improper code found in DUT RBUF after reset (skip selftest).
026346  012737  000622  003770  6$:       MOV     #0402.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE
026354  012701  007321                    MOV     #EM0402,R1      ;SELECT ERROR MESSAGE.
026360                                   ERROR                   ;REPORT ERROR.          >>>>> ERROR #0402 <<<<<
```

```
      026360  104460                                                          TRAP    C$ERROR
5218
5219 026362  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5220
5221 026366                                  ENDTST
      026366                                                          L10031:
      026366  104401                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RXFFDA -
                                ;++ ******************************************************************************
                                ;*              - RBUF Register RX Flag Field Test -
                                ;*      This test verifies that the Field of 3 flag bits in the RBUF reads
                                ;*      as all ones when the Selftest codes are being read from the DUT
                                ;*      after a board reset and skip selftest sequence.
                                ;*
                                ;-- ******************************************************************************
                                        BGNTST
                                                                                        T5::
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026370  012737  000005  002326         MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (5)
026376  012737  177777  002352         MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
026404  012737  000001  003766         MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026412  012737  007471  003772         MOV     #EM0501,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026420  012737  016406  003774         MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Set the DUT CSR Master Reset (MR) bit, perform the skip selftest sequence,
                                ;  and wait up to 5 seconds for the MR bit to clear.
                                ;-
026426  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
026432  012702  000040                  MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
026436  005003                          CLR     R3              ;WAITING FOR BIT TO CLEAR.
026440  013704  002244                  MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
026444  010214                          MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
026446  004737  022330                  JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
026452  004737  020340                  JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026456  103013                          BCC     4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; Read 8 characters from the DUT and verify that all 3 RX error flags are
                                ;  set for each characters.
                                ;-
026460  012400                          MOV     (R4)+,R0        ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
026462  012701  000010                  MOV     #8.,R1          ;INITIALIZE THE LOOP COUNTER.
026466  011402                  2$:     MOV     (R4),R2         ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
026470  012700  070000                  MOV     #70000,R0
026474  040200                          BIC     R2,R0           ;CALCULATE BIT MAP OF CLEAR RX ERROR FLAGS.
026476  001012                          BNE     6$              ;GO REPORT ERROR IF NOT ALL RX ERROR FLAGS SET.
026500  005301                          DEC     R1              ;COUNT THIS LOOP ITERATION.
026502  001371                          BNE     2$              ;LOOP IF NOT ALL LINES DONE.
026504  000415                          BR      60$             ;EXIT TEST, NO ERROR FOUND.
                                ;+
                                ; Error reports:
                                ;
                                    ;Report MR bit would not clear after a DUT reset.
026506  012737  000765  003770  4$:      MOV     #0501.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026514  012701  006173                  MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
026520                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0501 <<<<<
026520  104460                                                                                  TRAP    C$ERROR
026522  000406                          BR      60$             ;EXIT THE TEST.

                                    ;Report one or more RX error flags found set w th selftest code.
026524  012737  000766  003770  6$:      MOV     #0502.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026532  012701  007537                  MOV     #EM0502,R1      ;SELECT ERROR MESSAGE.
026536                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0502 <<<<<
026536  104460                                                                                  TRAP    C$ERROR
```

HARDWARE TEST - RXFFDA -

```
5277
5278 026540  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5279
5280 026544                                  ENDTST
     026544                                                                  L10032:
     026544  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RDAA -
                                ;++ ******************************************************************************
                                ;*              - CSR RX Data Available Bit Test -
                                ;*      This test verifies that the DUT CSR RX Data Available Bit is set by the
                                ;*      inclusion of the selftest codes in the DUT FIFO and that the bit clears
                                ;*      after the FIFO has been emptied.
                                ;*
                                ;-- ******************************************************************************
                                        BGNTST
                                                                                        T6::
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026546  012737  000006  002326         MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (6)
026554  012737  177777  002352         MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
026562  012737  000001  003766         MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026570  012737  010073  003772         MOV     #EM0601,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026576  012737  016406  003774         MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Set the DUT CSR Master Reset (MR) bit, perform the skip selftest sequence,
                                ;  and wait up to 5 seconds for the MR bit to clear.
                                ;-
026604  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
026610  012702  000040                  MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
026614  005003                         CLR     R3              ;WAITING FOR BIT TO CLEAR.
026616  013704  002244                 MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
026622  010214                         MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
026624  004737  022330                JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
026630  004737  020340                JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026634  103016                       BCC     4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; Check that the RX Data Available bit is set.
                                ;-
026636  032714  000200                  BIT     #BIT7,(R4)      ;TEST THE DUT RX.DATA.AVAIL BIT.
026642  001422                         BEQ     6$              ;GO REPORT ERROR IF BIT IS NOT SET.
                                ;+
                                ; Read characters from the DUT RX FIFO and wait for RX.DATA.AVAIL to go clear.
                                ;-
026644  012705  001130                  MOV     #600.,R5        ;ALLOW READING 600 CHARS BEFORE ERROR.
026650  010403                         MOV     R4,R3
026652  012300                         MOV     (R3)+,R0        ;CALCULATE THE RBUF ADDRESS.
026654  011300                 2$:     MOV     (R3),R0         ;READ A CHARACTER FROM THE RX FIFO.
026656  032714  000200                  BIT     #BIT7,(R4)      ;TEST THE DUT RX.DATA.AVAIL BIT.
026662  001427                         BEQ     60$             ;EXIT TEST WITHOUT ERROR IF RX.DATA.AVAIL CLR.
026664  005305                          DEC     R5              ;COUNT THE CHARACTER JUST READ.
026666  001372                         BNE     2$              ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
026670  000416                          BR      8$              ;GO REPORT ERROR IF RX.DATA.AVAIL WOULDN'T CLR.

                                ;+
                                ; Error reports:
                                ;-
                                    ;Report MR bit would not clear after a DUT reset.
026672  012737  001131  003770  4$:      MOV     #0601.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026700  012701  006173                  MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
026704                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0601 <<<<<
026704  104460                                                                                  TRAP    C$ERROR
026706  000415                          BR      60$             ;EXIT THE TEST.

                                    ;Report that RX.DATA.AVAIL bit was not set after a reset completion.
```

```
5337 026710  012737  001132  003770  6$:     MOV     #0602.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5338 026716  012701  010127                  MOV     #EM0602,R1      ;SELECT ERROR MESSAGE.
5339 026722                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0602 <<<<<
     026722  104400                                                                              TRAP    C$ERROR
5340 026724  000406                         BR      60$             ;EXIT THE TEST.
5341
5342                                 ;Report that RX.DATA.AVAIL bit could not be cleared by purging FIFO.
5343 026726  012737  001133  003770  8$:     MOV     #0603.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5344 026734  012701  010307                  MOV     #EM0603,R1      ;SELECT ERROR MESSAGE.
5345 026740                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0603 <<<<<
     026740  104460                                                                              TRAP    C$ERROR
5346
5347 026742  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5348
5349 026746                                  ENDTST
     026746                                                                      L10033:
     026746  104401                                                                          TRAP    C$ETST
```

HARDWARE TEST - RDVA -

```
                                .SBTTL  HARDWARE TEST           - RDVA -
                                ;++ ******************************************************************************
                                ;*              - RBUF RX Data Valid Bit Test -
                                ;*      This test verifies that the DUT RBUF RX Data Valid Bit is set by the
                                ;*      inclusion of the selftest codes in the DUT FIFO and that the bit clears
                                ;*      after the FIFO has been emptied.
                                ;*
                                ;-- ******************************************************************************
026750                                  BGNTST
026750                                                                          T7::
        000007                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026750  012737  000007  002326           MOV    @TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (7)
026756  012737  177777  002352           MOV    @-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
026764  012737  000001  003766           MOV    @1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026772  012737  010472  003772           MOV    @EM0701,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
027000  012737  016406  003774           MOV    @ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Set the DUT CSR Master Reset (MR) bit, perform the skip selftest sequence,
                                ;  and wait up to 5 seconds for the MR bit to clear.
                                ;-
027006  012701  011610           MOV    @5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
027012  012702  000040           MOV    @BIT05,R2       ;WAITING FOR MASTER RESET BIT.
027016  005003                   CLR    R3              ;WAITING FOR BIT TO CLEAR.
027020  013704  002244           MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
027024  010214                   MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
027026  004737  022330           JSR    PC,SKPSTS       ;SKIP THE SELFTEST.
027032  004737  020340           JSR    PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
027036  103012                   BCC    4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; Check that the RX Data Valid bit is set.
                                ;-
027040  012400                   MOV    (R4)+,R0        ;INCREMENT POINTER TO PNT TO DUT RBUF REG.
027042  005714                   TST    (R4)            ;TEST THE DUT RX.DATA.VALID BIT.
027044  100016                   BPL    6$              ;GO REPORT ERROR IF BIT IS NOT SET.
                                ;+
                                ; Read characters from the DUT RX FIFO and wait for RX.DATA.VALID to go clear.
                                ;-
027046  012705  001130           MOV    @600.,R5        ;ALLOW READING 600 CHARS BEFORE ERROR.
027052  011400           2$:     MOV    (R4),R0         ;READ A CHARACTER FROM THE RX FIFO.
027054  100027                   BPL    60$             ;EXIT TEST WITHOUT ERROR IF BIT IS CLEAR.
027056  005305                   DEC    R5              ;COUNT THE CHARACTER JUST READ.
027060  001374                   BNE    2$              ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
027062  000416                   BR     8$              ;GO REPORT ERROR IF RX.DATA.VALID WOULDN'T CLR.
                                ;+
                                ; Error reports:
                                ;-
                                    ;Report MR bit would not clear after a DUT reset.
027064  012737  001275  003770  4$:     MOV    @0701.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
027072  012701  006173                   MOV    @EM0202,R1      ;SELECT ERROR MESSAGE.
027076                                   ERROR                  ;REPORT ERROR.          >>>>> ERROR @0701 <<<<<
027076  104460                                                                            TRAP    C$ERROR
027100  000415                           BR     60$             ;EXIT THE TEST.
                                    ;Report that RX.DATA.VALID bit was not set after a reset completion.
027102  012737  001276  003770  6$:     MOV    @0702.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
027110  012701  010527                   MOV    @EM0702,R1      ;SELECT ERROR MESSAGE.
```

HARDWARE TEST - RDVA - SEQ 0140

```
5406 027114                              ERROR                  ;REPORT ERROR.          >>>>> ERROR #0702 <<<<<
     027114  104460                                                                            TRAP    C$ERROR
5407 027116  000406                      BR      60$            ;EXIT THE TEST.
5408
5409                                  ;Report that RX.DATA.VALID bit could not be cleared by purging FIFO.
5410 027120  012737  001277  003770  8$:     MOV     #0703.,ERRNBR  ;SET THE ERROR NUMBER IN ERROR TABLE.
5411 027126  012701  010707                  MOV     #EM0703,R1     ;SELECT ERROR MESSAGE.
5412 027132                              ERROR                  ;REPORT ERROR.          >>>>> ERROR #0703 <<<<<
     027132  104460                                                                            TRAP    C$ERROR
5413
5414 027134  005037  002352          60$:    CLR     CTRLCF         ;INDICATE THAT WE COMPLETED THE TEST.
5415
5416 027140                              ENDTST
     027140                                                                          L10034:
     027140  104401                                                                            TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RLNA -
                                ;** ****************************************************************************
                                ;*              - RBUF RX Line Number Field Test -
                                ;*      This test verifies that the DUT RBUF RX Line Number field is working
                                ;*      correctly by utilizing the selftest codes which are put in the RX
                                ;*      FIFO after a board reset.
                                ;*
                                ;-- ****************************************************************************
027142                                  BGNTST
027142                                                                          T8::
                000010                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027142  012737  000010  002326           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (8)
027150  012737  177777  002352           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
027156  012737  000001  003766           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027164  012737  011072  003772           MOV     #EM0801,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;*
                                ; Set the DUT CSR Master Reset (MR) bit, perform the skip selftest sequence,
                                ;  and wait up to 5 seconds for the MR bit to clear.
                                ;-
027172  012701  011610           MOV     #5000.,R1      ;TIME OUT VALUE IS 5.0 SECONDS.
027176  012702  000040           MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
027202  005003                   CLR     R3             ;WAITING FOR BIT TO CLEAR.
027204  013704  002244           MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
027210  010214                   MOV     R2,(R4)        ;SET THE DUT MASTER RESET BIT.
027212  004737  022330           JSR     PC,SKPSTS      ;SKIP THE SELFTEST.
027216  004737  020340           JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
027222  103016                   BCC     4$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;*
                                ; Read characters from the DUT RX FIFO and verify that the line numbers are
                                ;  correct.
                                ; One character is read from the FIFO for each possible line on the DUT.
                                ;
027224  005001                    CLR     R1             ;CLEAR THE LINE COUNTER.
027226  012400                   MOV     (R4)+,R0       ;INCREMENT POINTER TO PNT TO THE DUT RBUF REG.
027230  011402           2$:     MOV     (R4),R2        ;READ A CHARACTER FROM THE DUT RX FIFO.
027232  010203                   MOV     R2,R3
027234  000303                   SWAB    R3
027236  042703  177760           BIC     #177760,R3     ;REMOVE ALL BUT LINE NUMBER BITS.
027242  020301                   CMP     R3,R1          ;COMPARE WITH EXPECTED LINE NUMBER.
027244  001017                   BNE     6$             ;GO REPORT ERROR IF LINE NUMBERS DON'T MATCH.
027246  005201                    INC     R1             ;INCREMENT THE EXPECTED LINE NUMBER.
027250  020127  000010           CMP     R1,#NUMLNS     ;COMPARE WITH NUMBER OF LINES ON DUT.
027254  001365                   BNE     2$             ;LOOP UNTIL CODES FOR ALL LINES ARE READ.
027256  000423                    BR      60$            ;EXIT TEST WITHOUT ERROR.

                                ;*
                                ; Error reports:
                                ;-
                                    ;Report MR bit would not clear after a DUT reset.
027260  012737  001441  003770   4$:      MOV     #0801.,ERRNBR  ;SET THE ERROR NUMBER IN ERROR TABLE.
027266  012737  016470  003774            MOV     #ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
027274  012701  006173                    MOV     #EM0202,R1     ;SELECT ERROR MESSAGE.
027300                                   ERROR                  ;REPORT ERROR.          >>>>> ERROR #0801 <<<<<
027300  104460                                                                                    TRAP    C$ERROR
027302  000411                           BR      60$            ;EXIT THE TEST.

                                    ;Report that RX Line Number field is wrong for selftest code.
```

HARDWARE TEST - RLNA -

```
 5473 027304  012737  001442  003770  6$:     MOV     #0802.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
 5474 027312  012737  016406  003774          MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
 5475 027320  012701  011132                  MOV     #EM0802,R1      ;SELECT ERROR MESSAGE.
 5476 027324                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0802 <<<<<
      027324  104460                                                                              TRAP    C$ERROR
 5477
 5478 027326  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5479
 5480 027332                                  ENDTST
      027332                                                                      L10035:
      027332  104401                                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - BMPCHK -
                                ;++ ******************************************************************************
                                ;*              - BMP check Test -
                                ;*      This test is used to verify that the DUT does not immediately fail
                                ;*      the on-board background-monitor program, and hence invalidate
                                ;*      succeeding tests.
                                ;*      This test looks for BMP codes in the fifo for a set period immediately
                                ;*      after the self-test is skipped.
                                ;*      Any BMP codes that are found are saved on the queue and are also
                                ;*      reported in this test.
                                ;*
                                ;-- ******************************************************************************
027334                                  BGNTST
027334                                                                          T9::
                 000011                  TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027334  012737  000011  002326           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (9)
027342  012737  177777  002352           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
027350  012737  000001  003766           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027356  012737  001605  003770           MOV    #0901.,ERRNBR   ;SET THE ERROR NUMBER.
                                ;+
                                ; Wait up to 3 seconds for the DUT Master Reset bit to clear.
                                ;  If time-out occurs, then exit this test.
                                ;-
027364  012701  005670                   MOV    #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
027370  012702  000040                   MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
027374  005003                           CLR    R3              ;WAITING FOR BIT TO CLEAR.
027376  013704  002244                   MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
027402  004737  020340                   JSR    PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
027406  103027                           BCC    50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
                                ;+
                                ; Reset the DUT, skip the self-test.
                                ;-
027410  010214                           MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
027412  004737  022330                   JSR    PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
                                ;+
                                ; Wait for Master reset to clear. Delay for 500 milli-secs before purging
                                ; the fifo.
                                ;-
027416  012704  000764                   MOV    #500.,R4        ;TIME-OUT VALUE IS 500 MILLI-SECONDS.
027422  004737  020300                   JSR    PC,DELAY        ;WAIT FOR BMP TO BEGIN EXECUTION.
027426  004737  020716                   JSR    PC,PUFIFO       ;PURGE THE FIFO, SAVING ANY BMP CODES.
027432  103015                           BCC    50$             ;ABORT THE TEST IF THE FIFO DID NOT CLEAR.
                                ;+
                                ; Report the error if any BMP codes were found.
                                ;-
027434  013702  002524                   MOV    BMPCQP,R2       ;GET THE CONTENTS OF THE POINTER TO THE BMP Q.
027440  012703  002526                   MOV    #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
027444  020203                           CMP    R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
027446  001414                           BEQ    60$             ;EXIT NO CODES IN THE QUEUE.
                                ;+
                                ; There is at least one BMP code in the queue. Report the error.
                                ;
                                        ;Report error BMP CODE FOUND IN TEST nn, BMP CODE:nnnnnn"
027450  012701  011236                   MOV    #EM0902,R1      ;PASS THE MESSAGE TO BE REPORTED.
027454                                  ERRDF   0901,EM0901,ER9301 ;           >>>>> ERROR #0901 <<<<<.
027454  104455                                                                  TRAP    C$ERDF
027456  001605                                                                  .WORD   901
```

HARDWARE TEST - BMPCHK -

```
        027460  011205                                                                  .WORD   EM0901
        027462  017450                                                                  .WORD   ER9301
  5536  027464  000405                          BR      60$
  5537
  5538  027466  012737  001606  003770  50$:     MOV     #902.,ERRNBR    ;SET >>>>> ERROR #0902 <<<<<.
  5539  027474  004737  022456                  JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
  5540
  5541  027500  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
  5542
  5543  027504                          ENDTST
        027504                                                          L10036:
        027504  104401                                                          TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE TEST           - SKSELF -
                                        ;** ************************************************************************
                                        ;*              - Skip Self-test Test -
                                        ;*      This test verifies that the DUT skips the self-test within the
                                        ;*      time allowed, and that the fifo contains the correct codes after its
                                        ;*      completion.
                                        ;*
                                        ;-- ************************************************************************
                                            BGNTST
                                                                                        T10::
                                             TNUM == TNUM + 1           ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
                                             MOV    @TNUM,TSTNUM        ;SET UP THE TEST NUMBER.        (10)
                                             MOV    @-1,CTRLCF          ;INDICATE THAT WE ARE WITHIN A TEST.
                                             MOV    @1,ERRTYP           ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
                                             MOV    @EM1001,ERRMSG      ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                             MOV    @ER0503,ERRBLK      ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                        ;+
                                        ; Wait up to 3 seconds for the DUT Master Reset bit to clear.
                                        ;  If time-out occurs, then exit this test.
                                        ;
                                             MOV    @3000.,R1           ;TIME-OUT VALUE IS 3.0 SECONDS.
                                             MOV    @BIT05,R2           ;WAITING FOR MASTER RESET BIT.
                                             CLR    R3                  ;WAITING FOR BIT TO CLEAR.
                                             MOV    CSRA,R4             ;BIT IS IN THE DUT'S CSR.
                                            JSR     PC,MSLGET           ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
                                             BCC    50$                 ;ABORT THE TEST IF MR DID NOT CLEAR.
                                        ;+
                                        ; Determine if the DUT takes too short or too long a time to skip the self-test
                                        ;  Set-up a time-out of 50 milli-second, if MR is clear in less than 10 milli
                                        ;   -second, or greater than 50 milli-seconds, report the error.
                                        ;-
                                             MOV    @50.,R1             ;TIME-OUT VALUE IS 50 MILLI-SECONDS.
                                             MOV    R2,(R4)             ;SET THE DUT MASTER RESET BIT.
                                            JSR     PC,SKPSTS           ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
                                            JSR     PC,MSLGET           ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
                                            BCC     2$                  ;GO REPORT ERR IF SKIPPING STEST TOOK TOO LONG.
                                             CMP    R1,@40.
                                            BGT     4$                  ;GO REP ERR IF SELFTEST COMPLETED IN < 10 MS.
                                        ;+
                                        ; Self test completed within 10 milli-sec to 50 milli-seconds.
                                        ;  Verify that the self-test codes in the fifo are "good" codes ,ie the DUT
                                        ;  successfully completed the self-test.
                                        ; Th's subroutine reports errors with numbers >>>>> 1003 thru 1007 <<<<<.
                                        ;-
                                             MOV    @1003.,ERRNBR       ;SET ERROR NUMBER TO 1003.
                                            JSR     PC,RSTRPT           ;CHECK SELF-TEST CODES IN THE FIFO.
                                            BR      60$                 ;EXIT TEST.
                                        ;+
                                        ; Error reports:
                                        ;-
                                            ;Report Skip Self-test took too long.
                                        2$:  MOV    @1001.,ERRNBR       ;SET THE ERROR NUMBER IN THE ERROR TABLE.
                                             MOV    @EM1002,R1          ;SELECT ERROR MESSAGE.
                                            ERROR                       ;REPORT ERROR.          >>>>> ERROR @1001 <<<<<
                                                                                                    TRAP    C$ERROR
```

HARDWARE TEST - SKSELF - SEQ 0146

```
5600 027646  000414                          BR      60$             ;EXIT THE TEST.
5601
5602                                    ;Report Skip Self-test completed too soon.
5603 027650  012737  001752  003770  4 :     MOV     #1002.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5604 027656  012701  011423                  MOV     #EM1003,R1      ;SELECT ERROR MESSAGE.
5605 027662                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #1002 <<<<<
     027662  104460                                                                              TRAP    C$ERROR
5606 027664  000405                          BR      60$             ;EXIT THE TEST.
5607
5608 027666  012737  001753  003770  50$:    MOV     #1003.,ERRNBR   ;SET ERROR NUMBER.
5609 027674  004737  022456                  JSR     PC,TSABRT       ;REPORT NON TEST RELATED ERROR.
5610
5611 027700  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5612
5613 027704                                  ENDTST
     027704                                                                      L10037:
     027704  104401                                                                      TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - DFSKST -
                                ;++ ******************************************************************************
                                ;*              - Diagnostic fail bit, skip self-test test -
                                ;*      This test verifies that the diagnostic fail bit of the DUT, correctly
                                ;*      changes state as the on-boared selftest is skipped.
                                ;*
                                ;-- ******************************************************************************
                                        BGNTST
                                                                                T11::
                                 TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027706 012737 000013 002326      MOV    @TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (11)
027714 012737 177777 002352      MOV    @-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
027722 012737 000001 003766      MOV    @1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027730 012737 011501 003772      MOV    @EM1101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
027736 012737 016470 003774      MOV    @ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Wait up to 3 seconds for the DUT Master Reset bit to clear.
                                ;  If time-out occurs, then exit this test.
                                ;-
027744 012701 005670             MOV    @3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
027750 012702 000040             MOV    @BIT05,R2       ;WAITING FOR MASTER RESET BIT.
027754 005003                    CLR    R3              ;WAITING FOR BIT TO CLEAR.
027756 013704 002244             MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
027762 004737 020340            JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
027766 103044                    BCC    50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
                                ;+
                                ; Reset the DUT, skip the self-test.
                                ;-
027770 010214                    MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
027772 004737 022330            JSR     PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
                                ;+
                                ; Set time out of 5 milli seconds, wait for Diag_fail bit to set.
                                ;  If time out occurs go report the error.
                                ;-
027776 012701 000005             MOV    @5,R1           ;TIME-OUT VALUE IS 5 MILLI-SECONDS.
030002 012702 020000             MOV    @BIT13,R2       ;WAITING FOR DIAGNOSTIC FAIL BIT.
030006 010203                    MOV    R2,R3           ;WAITING FOR BIT TO SET.
030010 013704 002244             MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
030014 004737 020340            JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
030020 103020                   BCC     4$              ;IF DIAG_FAIL DID NOT SET, GO REPORT ERROR.

                                ;+
                                ; Set time-out of 15 milli-secs, wait for Diag_Fail to clear.
                                ;  If time-out occurs go report the error.
                                ;  Verify the Diag_fail bit is in a stable state before continuing.  Loop
                                ;  back if the state was transitory, using the remainder of the 15 ms time out.
                                ;-
030022 012701 000017             MOV    @15.,R1         ;TIME-OUT VALUE IS 15 MILLI-SECONDS.
030026 005003           2$:      CLR    R3              ;WAITING FOR BIT TO CLEAR.
030030 004737 020340            JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
030034 103012                    BCC    4$              ;IF DIAG_FAIL DID NOT CLEAR, GO REPORT ERROR.
030036 010105                    MOV    R1,R5           ;SAVE THE REMAINING TIME-OUT VALUE.
030040 012701 000001             MOV    @1,R1           ;SET TIME-OUT OF 1 MILLI-SECOND.
030044 052703 020000             BIS    @BIT13,R3       ;WAIT FOR BIT TO SET.
030050 004737 020340            JSR     PC,MSLGET       ;DOUBLE CHECK TO ELIMINATE NOISE PROBLEMS.
```

HARDWARE TEST - DFSKST - SEQ 0148

```
 5671 030054  103016                    BCC     60$             ;EXIT IF DIAG_FAIL BIT STILL CLEAR.
 5672 030056  010501                     MOV     R5,R1           ;PASS THE REMAINING TIME-OUT VALUE.
 5673 030060  000762                    BR      2$              ;LOOP TO CHECK AGAIN.
 5674                            ;*
 5675                            ; Error reports:
 5676                            ;-
 5677                                ;Report Diagnostic fail bit bad.
 5678 030062  012737  002115  003770  4$:      MOV     #1101.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
 5679 030070  012701  011742                    MOV     #EM1205,R1      ;SELECT ERROR MESSAGE.
 5680 030074                            ERROR                   ;REPORT ERROR.          >>>>> ERROR #1101 <<<<<
      030074  104460                                                                            TRAP    C$ERROR
 5681 030076  000405                    BR      60$             ;EXIT THE TEST.
 5682
 5683 030100  012737  002116  003770  50$:     MOV     #1102.,ERRNBR   ;SET THE ERROR NUMBER FOR TSABRT RTN.
 5684 030106  004737  022456                    JSR     PC,TSABRT       ;REPORT NON TEST RELATED ERROR.
 5685
 5686 030112  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED A TEST.
 5687
 5688 030116                            ENDTST
      030116                                                                    L10040:
      030116  104401                                                                TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - SELFTS -
                                ;++ ******************************************************************************
                                ;*              - Self-test Test -
                                ;*      This test verifies that the DUT's Self-Test executes within the
                                ;*      time allowed, and that the fifo contains the correct codes after its
                                ;*      completion.
                                ;*
                                ;-- ******************************************************************************
                                        BGNTST
                                                                                        T12::
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
030120  012737  000014  002326         MOV     @TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (12)
030126  012737  177777  002352         MOV     @-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
030134  012737  000001  003766         MOV     @1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
030142  012737  011525  003772         MOV     @EM1201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
030150  012737  016470  003774         MOV     @ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Wait up to 3 seconds for the DUT Master Reset bit to clear.
                                ;  If time-out occurs, then exit this test.
                                ;-
030156  012701  005670                 MOV     @3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
030162  012702  000040                 MOV     @BIT05,R2       ;WAITING FOR MASTER RESET BIT.
030166  005003                         CLR     R3              ;WAITING FOR BIT TO CLEAR.
030170  013704  002244                 MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
030174  004737  020340                 JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030200  103062                         BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
                                ;+
                                ; Determine if the Self-test takes too short or too long a time to complete.
                                ;  Set-up a time-out of 3 second, if MR is clear in less than 1/2 second, or
                                ;  greater than 3 seconds, report the error.
                                ;
030202  012701  005670                 MOV     @3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
030206  010214                         MOV     R2,(R4)         ;SET THE DUT MASTER RESE" BIT.
030210  004737  020340                 JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030214  103030                         BCC     4$              ;GO REPORT ERROR SELFTEST TOOK TOO LONG.
030216  012702  005670                 MOV     @3000.,R2
030222  160102                         SUB     R1,R2           ;CALCULATE @ OF MS SELFTEST TO COMPLETE.
030224  020227  000062                 CMP     R2,@50.
030230  002431                         BLT     6$              ;SELFTEST SKIPPED?  YES, GO REPORT ERROR.
030232  020227  000764                 CMP     R2,@500.
030236  002434                         BLT     8$              ;GO REP ERR IF SELFTEST COMPLETED IN < 1/2 SEC.
                                ;+
                                ; Self-test completed within 1sec to 3 seconds.
                                ;  Check the state of the Diagnostic Fail bit, report error if it is set.
                                ;-
030240  032714  020000                 BIT     @BIT13,(R4)     ;DETERMINE IF THE DIAG_FAIL BIT IS CLEAR.
030244  001406                         BEQ     2$              ;SKIP ERROR REPORT IF BIT IS CLEAR.
                                    ;Report Diagnostic fail bit bad.
030246  012737  002264  003770         MOV     @1204.,ERRNBR   ;SET ERROR NUMBER TO IN ERROR TABLE.
030254  012701  011742                 MOV     @EM1205,R1      ;SELECT THE ERROR MESSAGE.
030260                                 ERROR                   ;               >>>>> ERROR @1204 <<<<<
030260  104460                                                                           TRAP    C$ERROR
                                ;+
                                ; Verify that the self-test codes in the fifo are "good" codes ,ie the DUT
                                ; successfully completed the self-test.
                                ; This subroutine reports errors with numbers >>>>> 1205 thru 1209 <<<<<.
                                ;-
```

HARDWARE TEST - SELFTS -

```
 5745 030262  012737  002265  003770  2$:     MOV     #1205.,ERRNBR   ;SET ERROR NUMBER TO 1205.
 5746 030270  004737  021622                  JSR     PC,RSTRPT       ;CHECK SELF-TEST CODES IN THE FIFO.
 5747 030274  000431                          BR      60$             ;EXIT TEST.
 5748                                 ;+
 5749                                 ; Error reports:
 5750                                 ;-
 5751                                     ;Report Self-test took too long to complete.
 5752 030276  012737  002261  003770  4$:     MOV     #1201.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
 5753 030304  012701  011544                  MOV     #EM1202,R1      ;SELECT ERROR MESSAGE.
 5754 030310                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #1201 <<<<<
      030310  104460                                                                              TRAP    C$ERROR
 5755 030312  000422                          BR      60$             ;EXIT THE TEST.
 5756
 5757                                     ;Report Self-test did not execute after DUT reset.
 5758 030314  012737  002262  003770  6$:     MOV     #1202.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
 5759 030322  012701  011706                  MOV     #EM1204,R1      ;SELECT ERROR MESSAGE.
 5760 030326                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #1202 <<<<<
      030326  104460                                                                              TRAP    C$ERROR
 5761
 5762                                     ;Report Self-test competed too soon.
 5763 030330  012737  002263  003770  8$:     MOV     #1203.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
 5764 030336  012701  011630                  MOV     #EM1203,R1      ;SELECT ERROR MESSAGE.
 5765 030342                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #1203 <<<<<
      030342  104460                                                                              TRAP    C$ERROR
 5766 030344  000405                          BR      60$             ;EXIT THE TEST.
 5767
 5768 030346  012737  002272  003770  50$:    MOV     #1210.,ERRNBR   ;SET THE ERROR NUMBER FOR TSABRT RTN.
 5769 030354  004737  022456                  JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
 5770
 5771 030360  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5772
 5773 030364                                  ENDTST
      030364                                                                          L10041:
      030364  104401                                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - STFAIL -
                                ;++ ******************************************************************************
                                ;*              - Self-test fail test -
                                ;*      This test verifies that the DUT will report selftest errors via the
                                ;*      fifo.  And that the Diagnostic fail bit will indicate the error.
                                ;*      This is accomplished via a software 'hook' in the self-test, which
                                ;*      forces a "Proc1 to ram error" to be placed in the fifo.
                                ;*
                                ;-- ******************************************************************************
030366                                  BGNTST
030366                                                                          T13::
000015                                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
030366  012737  000015  002326           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (13)
030374  012737  177777  002352           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
030402  012737  000001  003766           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
030410  012737  011766  003772           MOV    #EM1301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
030416  012737  016470  003774           MOV    #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
030424  012737  002425  003770           MOV    #1301.,ERRNBR   ;SET ERROR NUMBER TO 1301.
                                ;+
                                ; Wait up to 3 seconds for the DUT Master Reset bit to clear.
                                ;  If time-out occurs, then exit this test.
                                ;-
030432  012701  005670                   MOV    #3000.,R1       ;TIME OUT VALUE IS 3.0 SECONDS.
030436  012702  000040                   MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
030442  005003                           CLR    R3              ;WAITING FOR BIT TO CLEAR.
030444  013704  002244                   MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
030450  004737  020340                   JSR    PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030454  103120                           BCC    50$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; Reset the DUT, check for rom version 0.
                                ;  If Version 0 is found then exit this test.
                                ;-
030456  010214                           MOV    R2,(R4)         ;RESET THE DUT.
030460  004737  022330                   JSR    PC,SKPSTS       ;SKIP THE SELFTEST.
030464  012701  000764                   MOV    #500.,R1        ;PASS TIME-OUT VALUE OF 500 MILLISECS.
030470  004737  020340                   JSR    PC,MSLGET       ;WAIT FOR MR BIT TO CLEAR.
030474  103110                           BCC    50$             ;GO REPORT ERROR IF TIME-OUT OCCURRED.
030476  005237  003770                   INC    ERRNBR          ;SET ERROR NUMBER TO 1302.
030502  012700  000010                   MOV    #8.,R0          ;SET MAXIMUM READ COUNT.
030506  005003                           CLR    R3              ;CLEAR THE ROM VERSION 0 FLAG.
030510  013704  002246                   MOV    RBUFA,R4        ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
030514  011402                   2$:     MOV    (R4),R2         ;READ A CODE FROM THE FIFO.
030516  100077                           BPL    50$             ;GO REPORT ERROR IF THE FIFO IS EMPTY.
030520  042702  007402                   BIC    #7402,R2        ;REMOVE THE LINE NUMBER AND PROC INDICATOR.
030524  020227  170001                   CMP    R2,#170001      ;COMPARE WITH ROM VERSION #0 CODE.
030530  001002                           BNE    4$              ;ROM VERSION #0?  NO, SKIP SETTING FLAG.
030532  012703  177777                   MOV    #-1,R3          ;YES, SET THE ROM VERSION #0 FLAGS.
030536  005300                   4$:     DEC    R0              ;DECREMENT MAX READ COUNTER.
030540  001365                           BNE    2$              ;LOOP IF 8 CODES HAVE NOT BEEN READ.
030542  005703                           TST    R3              ;CHECK THE ROM VERSION #1 INDICATOR.
030544  100466                           BMI    60$             ;ROM VERSION 0 IN EITHER PROCESSOR?  YES, EXIT.

                                ;+
                                ; Reset the DUT, delay for 25 milli-seconds before writing the Fail_self_test
                                ; code to TBUFFCT register on channel 0.
                                ;-
```

```
 5831 030546  012777  000040  151470          MOV     #BIT05,@CSRA    ;SET DUT MASTER RESET BIT, SELECT CHANNEL 0.
 5832 030554  012704  000031                  MOV     #25.,R4         ;PASS DELAY PERIOD OF 25 MILLI SECS.
 5833 030560  004737  020300                  JSR     PC,DELAY        ;WAIT FOR SELFTEST TO INITIALISE.
 5834 030564  012777  146314  151470          MOV     #146314,@TXBFCA ;WRITE THE FAIL SELF-TEST CODE TO TBUFFCT REG.
 5835                                ;+
 5836                                ; Wait up to 2 seconds for the self-test to complete.
 5837                                ;  If time-out occurs, then exit this test.
 5838                                ;-
 5839 030572  005237  003770                  INC     ERRNBR          ;SET ERROR NUMBER TO 1303.
 5840 030576  012701  003720                  MOV     #2000.,R1       ;TIME-OUT VALUE IS 2.0 SECONDS.
 5841 030602  012702  000040                  MOV     #BIT05,R2       ;PASS THE BIT MAP OF THE BIT TO TEST.
 5842 030606  005003                          CLR     R3              ;SET UP THE EXPECTED STATE.
 5843 030610  013704  002244                  MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
 5844 030614  004737  020340                  JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
 5845 030620  103036                          BCC     50$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
 5846                                ;+
 5847                                ; Verify the diagnostic fail bit is set, indicating the error.
 5848                                ; Report error if diagnostic fail bit is clear.
 5849                                ;-
 5850 030622  005237  003770                  INC     ERRNBR          ;SET ERROR NUMBER TO 1304.
 5851 030626  032714  020000                  BIT     #BIT13,(R4)     ;CHECK THE STATE OF THE DIAG_FAIL BIT.
 5852 030632  001425                          BEQ     10$             ;GO REPORT ERROR IF DIAG_FAIL BIT CLEAR.
 5853                                ;+
 5854                                ; Remove the 8 self-test codes form the fifo, and verify that at least
 5855                                ; one is a Proc1 to ram error code (231).
 5856                                ;-
 5857 030634  005237  003770                  INC     ERRNBR          ;SET ERROR NUMBER TO 1305.
 5858 030640  012700  000010                  MOV     #8.,R0          ;SET MAXIMUM READ COUNT.
 5859 030644  005001                          CLR     R1              ;CLEAR THE CORRECT CODE COUNTER.
 5860 030646  013704  002246                  MOV     RBUFA,R4        ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
 5861 030652  011402                  6$:     MOV     (R4),R2         ;READ A CODE FROM THE FIFO.
 5862 030654  100020                          BPL     50$             ;GO REPORT ERROR IF THE FIFO IS EMPTY.
 5863 030656  042702  007400                  BIC     #7400,R2        ;REMOVE THE LINE NUMBER FROM THE CODE.
 5864 030662  120227  170231                  CMPB    R2,#170231      ;IS IT THE CORRECT ERROR CODE?.
 5865 030666  001001                          BNE     8$              ;SKIP NEXT INSTRUCTION, IF NOT A 231 CODE.
 5866 030670  005201                          INC     R1              ;INCREMENT COUNTER.
 5867 030672  005300                  8$:     DEC     R0              ;DECREMENT MAX READ COUNTER.
 5868 030674  001366                          BNE     6$              ;LOOP IF 8 CODES HAVE NOT BEEN READ.
 5869 030676  005701                          TST     R1              ;WERE ANY 231 CODES FOUND?.
 5870 030700  001010                          BNE     60$             ;YES, THEN EXIT.
 5871 030702  005237  003770                  INC     ERRNBR          ;NO, SET ERROR NUMBER TO 1306 AND REPORT ERROR.
 5872                            ;Report Self-test error reporting bad.
 5873 030706  012701  012012          10$:    MOV     #EM1302,R1      ;SELECT ERROR MESSAGE.
 5874 030712                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR <<<<<
      030712  104460                                                                              TRAP    C$ERROR
 5875 030714  000402                          BR      60$             ;EXIT THE TEST.
 5876
 5877 030716  004737  022456          50$:    JSR     PC,TSABRT       ;REPORT NON RELATED TEST ERROR.
 5878
 5879 030722  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5880
 5881 030726                                  ENDTST
      030726                                                                      L10042:
      030726  104401                                                                      TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE TEST           - ROMVER -
                                        ;++ ******************************************************************************
                                        ;*              - Rom Version Test -
                                        ;*      This test verifies that the DUT's Self-Test places valid Rom version
                                        ;*      numbers in the fifo after it has been skipped.  The Rom version numbers
                                        ;*      will be reported (on the first pass only), if an affirmative answer
                                        ;*      was given to the software P-table question.
                                        ;*
                                        ;-- ******************************************************************************
030730                                          BGNTST
030730                                                                                  T14::
                000016                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
030730  012737  000016  002326              MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (14)
030736  012737  177777  002352              MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
030744  012737  000001  003766              MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
030752  012737  012051  003772              MOV     #EM1401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
030760  012737  016470  003774              MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                        ;+
                                        ; Wait up to 3 seconds for the DUT Master Reset bit to clear.
                                        ;  If time-out occurs, then exit this test.
                                        ;-
030766  012701  005670                          MOV     #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
030772  012702  000040                          MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
030776  005003                                  CLR     R3              ;WAITING FOR BIT TO CLEAR.
031000  013704  002244                          MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
031004  004737  020340                          JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
031010  103131                                  BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
                                        ;+
                                        ; Set the Master Reset bit, and skip the self test.
                                        ;-
031012  010214                                  MOV     R2,(R4)         ;SET THE MASTER RESET BIT.
031014  004737  022330                          JSR     PC,SKPSTS       ;SKIP THE SELF TEST.
031020  012701  005670                          MOV     #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
031024  004737  020340                          JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
031030  103121                                  BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
                                        ;+
                                        ; Remove characters from the FIFO until either;
                                        ; (a) The FIFO is purged, go report the error.
                                        ; (b) The maximum try counter is zero, go report the error.
                                        ; (c) PROC_1's rom version number was found before PROC_2's, go report error.
                                        ; (d) Both ROM version numbers have been found.
                                        ;-
031032  012705  000040                          MOV     #4*NUMLNS,R5    ;SET MAXIMUM TRY COUNTER.
031036  012703  000143                          MOV     #99.,R3         ;SET AN INVALID ROM VERSION NUMBER FOR PROC_1.
031042  010304                                  MOV     R3,R4           ;SET AN INVALID ROM VERSION NUMBER FOR PROC_2.
031044  012737  002571  003770                  MOV     #1401.,ERRNBR   ;SET THE ERROR NUMBER TO 1401.
031052  012701  012101                          MOV     #EM1402,R1      ;SELECT MESSAGE TO BE REPORTED IF FIFO EMPTY.

031056  017702  151164                  2$:     MOV     @RBUFA,R2       ;READ THE NEXT CHAR FROM THE FIFO.
031062  100077                                  BPL     12$             ;GO REPORT ERROR IF FIFO EMPTY.
                                          ;+
                                          ; Check if the read data is a BMP code.
                                          ;-
031064  012700  000301                          MOV     #301,R0         ;SET UP A BIT MASK OF A BMP CODE.
031070  040200                                  BIC     R2,R0           ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
```

```
5939 031072  001003                    BNE     4$              ;BRANCH IF NOT A BMP CODE.
5940 031074  004737  022262            JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
5941 031100  000435                    BR      8$              ;
5942                                 ;+
5943                                 ; Check if the read data 's a self-test code.
5944                                 ;
5945 031102  012700  000201        4$:  MOV     #201,R0         ;SET-UP A BIT MASK OF A SELFTEST CODE.
5946 031106  040200                    BIC     R2,R0           ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
5947 031110  001431                   BEQ     8$              ;BRANCH IF IT IS A SELFTEST CODE.
5948
5949                                 ;+
5950                                 ; The read data is a ROM version number, determine which one it is.
5951                                 ;-
5952 031112  032702  000002            BIT     #BIT1,R2        ;CHECK THE PROCESSOR NUMBER BIT IN THE CODE.
5953 031116  001407                   BEQ     6$              ;BRANCH IF IT IS PROC_1 ROM VERSION NUMBER.
5954 031120  010204                    MOV     R2,R4           ;SAVE PROC_2 ROM VERSION NUMBER.
5955 031122  042704  177603            BIC     #177603,R4      ;CLEAR ANY UNWANTED BITS.
5956 031126  000241                    CLC                     ;CLEAR THE CARRY BIT.
5957 031130  006004                    ROR     R4              ;SHIFT THE CODES ALONG TO GET THE ROM
5958 031132  006004                    ROR     R4              ; VERSION NUMBER IN THE LOW 5 BITS.
5959 031134  000417                   BR      8$              ;
5960 031136  010203                6$:  MOV     R2,R3           ;SAVE PROC_1 ROM VERSION NUMBER.
5961 031140  042703  177603            BIC     #177603,R3      ;CLEAR ANY UNWANTED BITS.
5962 031144  000241                    CLC                     ;CLEAR THE CARRY BIT.
5963 031146  006003                    ROR     R3              ;SHIFT THE CODE ALONG TO GET THE ROM
5964 031150  006003                    ROR     R3              ; VERSION NUMBER IN THE LOW 5 BITS.
5965 031152  020427  000143            CMP     R4,#99.         ;CHECK IF WE HAVE RECEIVE PROC_2 ROM CODE.
5966 031156  001016                   BNE     10$             ;GO REPORT BOTH ROM VERSION NUMBERS.
5967                                 ;+
5968                                 ; Received ROM version numbers out of sequence.
5969                                 ;  ie, Proc_1's ROM version number found in the fifo before Proc_2's.
5970                                 ;-
5971 031160  012701  012167            MOV     #EM1403,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5972 031164  012737  002572  003770    MOV     #1402.,ERRNBR   ;SET THE ERROR NUMBER.
5973 031172  000433                   BR      12$             ;GO REPORT ERROR.
5974
5975 031174  005305                6$:  DEC     R5              ;DECREMENT THE MAX TRY COUNTER.
5976 031176  001327                   BNE     2$              ;LOOP TO GET THE NEXT CHAR FROM THE FIFO.
5977 031200  012701  012242            MOV     #EM1404,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5978 031204  012737  002573  003770    MOV     #1403.,ERRNBR   ;SET THE ERROR NUMBER.
5979 031212  000423                    BR      12$             ;GIVE UP, GO REPORT ERROR.
5980                                 ;+
5981                                 ; If this is the first pass, and software P-table question was answered YES,
5982                                 ; then report the rom version numbers to the operator.
5983                                 ;-
5984 031214  032737  000001  002226 10$:  BIT     #BIT0,OPTION    ;CHECK ON THE STATE OF THE SOFTWARE SWITCH.
5985 031222  001431                   BEQ     60$             ;EXIT IF NO ROM VERSION PRINTOUT WAS REQUESTED.
5986 031224  023727  002332  000001    CMP     PASCNT,#1       ;CHECK IF THIS IS THE FIRST PASS.
5987 031232  003025                   BGT     60$             ;EXIT IF ROM VERS HAVE ALREADY BEEN REPORTED.
5988 031234                            PRINTB #EF1401,R3,R4    ;PRINT THE ROM VERSION NUMBERS.
     031234  010446                                                    MOV     R4,-(SP)
     031236  010346                                                    MOV     R3,-(SP)
     031240  012746  004214                                            MOV     #EF1401,-(SP)
     031244  012746  000003                                            MOV     #3,-(SP)
     031250  010600                                                    MOV     SP,R0
     031252  104414                                                    TRAP    C$PNTB
     031254  062706  000010                                            ADD     #10,SP
```

HARDWARE TEST - ROMVER -

```
 5989 031260  000412                          BR      60$             ;EXIT THIS TEST.
 5990                                 ;-
 5991                                 ; Error reports:
 5992                                 ;-
 5993 031262  012737  016562  003774  12$:     MOV     #ER1401,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
 5994 031270                          ERROR                           ;REPORT ERROR.          >>>>> ERROR <<<<<
      031270  104460                                                                                TRAP    C$ERROR
 5995 031272  000405                          BR      60$
 5996
 5997 031274  012737  002575  003770  50$:     MOV     #1405.,ERRNBR   ;SET UP ERROR NUMBER FOR TSABRT RTN.
 5998 031302  004737  022456                  JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
 5999
 6000 031306  005037  002352          60$:     CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 6001
 6002 031312                          ENDTST
      031312                                                                          L10043:
      031312  104401                                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - REGWRW -
                                ;++ ******************************************************************************
                                ;*              - Device Register Word Access Read and Write Test -
                                ;*
                                ;* This test verifies that the device registers can be read and written
                                ;* correctly using word accesses.
                                ;*
                                ;-- ******************************************************************************

031314                                  BGNTST
031314                                                                          T15::
000017                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031314  012737  000017  002326      MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (16)
031322  012737  177777  002352      MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
031330  012737  000001  003766      MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
031336  012737  003101  003770      MOV     #1601.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
031344  012737  012445  003772      MOV     #EM1604,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
031352  005037  002462              CLR     ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
031356  012700  002464              MOV     #ERCNTB,R0
031362  004737  020200              JSR     PC,CLR16W       ;CLEAR THE ERROR COUNTER TABLE.
                                ;+
                                ; Reset the DUT to a known state, do not remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports errors >>>>> 1601 <<<<<.
                                ;-
031366  004737  021456              JSR     PC,RESETT       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
031372  103402                      BCS     .+6             ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
031374  000137  031516              JMP     60$             ;YES,  EXIT THE TEST.
                                ;+
                                ; Verify read/write capability to indirect address field of CSR
                                ;-
031400  005237  003770              INC     ERRNBR          ;SET THE ERROR REPORT NUMBER TO 1602.
031404  012702  000017              MOV     #17,R2          ;SET LOOP COUNT.
031410  013704  002244              MOV     CSRA,R4         ;GET CSR ADDRESS.
031414  010214                  2$: MOV     R2,(R4)         ;WRITE COUNT TO CSR.
031416  011401                      MOV     (R4),R1         ;READ BACK THE CONTENTS OF THE CSR
031420  042701  177760              BIC     #177760,R1      ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
031424  020102                      CMP     R1,R2           ;CHECK FOR CORRECT DATA WRITTEN/READ.
031426  001406                  BEQ     4$              ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                ;Report "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
031430  012737  016712  003774      MOV     #ER1601,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
031436  005003                      CLR     R3              ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
031440  005005                      CLR     R5              ;CAUSE REPORT OF LINE 0.
031442                          ERROR                       ;  >>>>> ERROR # 1602 <<<<<
031442  104460                                                                  TRAP    C$ERROR
031444  005302                  4$: DEC     R2              ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
031446  002362                  BGE     2$              ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
                                ;+
                                ; Write and verify 16 data patterns in all used bits of all registers on all
                                ;   active lines.  Before writing each pattern, clear all the bits.
                                ;    REGTST routine reports errors with numbers  >>>>> ERROR 1603  1605 <<<<<.
                                ;-
031450  005237  003770              INC     ERRNBR          ;SET THE ERROR NUMBER TO 1603.
031454  005003                      CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED
031456  012704  000002              MOV     #2,R4           ;INDICATE R/W ACCESS. CLEAR FIRST.
031462  004737  021220              JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS
                                ;+
```

```
                                        ; Write and verify 16 data patterns in all used bits of all registers on all
                                        ;   active lines.  Before writing each pattern, set all the bits.
                                        ;    REGTST routine reports errors with numbers  >>>>> ERROR 1606 - 1608 <<<<<.
                                        ;-
031466  012737  003106  003770                 MOV     #1606.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
031474  00500?                                 CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
031476  00540.                                 NEG     R4              ;INDICATE R/W ACCESS, SET FIRST.
031500  004737  021220                        JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                        ;-
                                        ; Print error summary reports if necessary.
                                        ;   The following routine reports errors with number >>>>> ERROR # 1609 <<<<<
                                        ;-
031504  012737  003111  003770                 MOV     #1609.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
031512  004737  021430                        JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
031516  005037  002352                60$:     CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
031522                                        ENDTST
031522                                                                                  L10044:
031522  104401                                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - REGWRM -
                                ;** ************************************************************************
                                ;*      - Device Register Word Access Read/Modify/Write Test -
                                ;*
                                ;* This test verifies that the device registers can be written correctly
                                ;* using word read/modify/write accesses.
                                ;*
                                ;-- ************************************************************************

                                        BGNTST
                                                                                        T16::
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031524  012737  000020  002326          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (17)
031532  012737  177777  002352          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
031540  012737  000001  003766          MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
031546  012737  003245  003770          MOV     #1701.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
031554  012737  012513  003772          MOV     #EM1701,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
031562  005037  002462                  CLR     ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
031566  012700  002464                  MOV     #ERCNT9,R0
031572  004737  020200                 JSR     PC,CLR16W       ;CLEAR THE ERROR COUNTER TABLE.
                                ;+
                                ; Reset the DUT to a known state, do not remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports errors >>>>> 1701 <<<<<.
                                ;-
031576  004737  021456                  JSR     PC,RESETT       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
031602  103402                         BCS     .+6             ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
031604  000137  031732                 JMP     60$             ;YES,  EXIT THE TEST.
                                ;+
                                ; Verify read/modify/write capability to indirect address field of CSR
                                ;-
031610  005237  003770                  INC     ERRNBR          ;SET THE ERROR REPORT NUMBER TO 1702.
031614  012702  000017                  MOV     #17,R2          ;SET LOOP COUNT.
031620  013704  002244                  MOV     CSRA,R4         ;GET CSR ADDRESS.
031624  042714  000017          2$:     BIC     #17,(R4)        ;CLEAR THE DUT CSR USING READ/MODIFY/WRITE.
031630  050214                          BIS     R2,(R4)         ;WRITE COUNT TO CSR USING READ/MODIFY/WRITE.
031632  011401                          MOV     (R4),R1         ;READ BACK THE CONTENTS OF THE CSR
031634  042701  177760                  BIC     #177760,R1      ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
031640  020102                          CMP     R1,R2           ;CHECK FOR CORRECT DATA WRITTEN/READ.
031642  001406                         BEQ     4$              ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                    ;Report "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
031644  012737  016712  003774          MOV     #ER1601,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
031652  005003                          CLR     R3              ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
031654  005005                          CLR     R5              ;CAUSE REPORT OF LINE 0.
031656                                 ERROR                   ;  >>>>> ERROR # 1702 <<<<<
031656  104460                                                                  TRAP    C$ERROR
031660  005302                  4$:     DEC     R2              ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
031662  002360                         BGE     2$              ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
                                ;+
                                ; Write and verify 16 data patterns in all used bits of all registers on all
                                ;   active lines using R/M/W.  Before writing each pattern, clear all the bits.
                                ;    REGTST routine reports errors with numbers  >>>>> ERROR 1703 - 1705 <<<<<.
                                ;-
031664  005237  003770                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 1703.
031670  005003                          CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
031672  012704  000001                  MOV     #1,R4           ;INDICATE R/M/W ACCESS, CLEAR FIRST.
031676  004737  021220                 JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
```

```
                                        ;*
                                        ; Write and verify 16 data patterns in all used bits of all registers on all
                                        ;   active lines using R/M/W.  Before writing each pattern, set all the bits.
                                        ;    REGTST routine reports errors with numbers  >>>>> ERROR 1706 - 1708 <<<<<.
                                        ;-
6136 031702  012737  003252  003770          MOV     #1706.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6137 031710  005003                          CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6138 031712  005404                          NEG     R4              ;INDICATE R/M/W ACCESS, SET FIRST.
6139 031714  004737  021220                 JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6140                                    ;*
6141                                    ; Print error summary reports if necessary.
6142                                    ;   The following routine reports errors with number >>>>> ERROR # 1709 <<<<<
6143                                    ;
6144 031720  012737  003255  003770          MOV     #1709.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
6145 031726  004737  021430                 JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
6146 031732  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
6147 031736                                 ENDTST
     031736                                                                     L10045:
     031736  104401                                                                 TRAP    C$ETST
```

```
 6149                                   .SBTTL  HARDWARE TEST           - REGBRW -
 6150                                   ;++ ************************************************************************
 6151                                   ;*              - Device Register Byte Access Read and Write Test -
 6152                                   ;*
 6153                                   ;* This test verifies that the device registers can be read and written
 6154                                   ;* correctly using byte accesses.
 6155                                   ;*
 6156                                   ;-- ************************************************************************
 6157
 6158 031740                                    BGNTST
      031740                                                                          T17::
 6159           000021                           TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 6160 031740   012737   000021   002326         MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (18)
 6161 031746   012737   177777   002352         MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WTTHIN A TEST.
 6162 031754   012737   000001   003766         MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
 6163 031762   012737   003411   003770         MOV     #1801.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
 6164 031770   012737   012570   003772         MOV     #EM1801,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
 6165 031776   005037   002462                 CLR     ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
 6166 032002   012700   002464                 MOV     #ERCNTB,R0
 6167 032006   004737   020200                JSR     PC,CLR16W       ;CLEAR THE ERROR COUNTER TABLE.
 6168                                   ;+
 6169                                   ; Reset the DUT to a known state, do not remove the status codes from the fifo.
 6170                                   ;  Clear TX and RX interrupt enable bits in the CSR.
 6171                                   ; This subroutine reports errors >>>>> 1801 <<<<<.
 6172                                   ;-
 6173 032012   004737   021456                 JSR     PC,RESETT       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
 6174 032016   103402                         BCS     .+6             ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
 6175 032020   000137   032176                 JMP     60$             ;YES,  EXIT THE TEST.
 6176 032024   012737   003412   003770         MOV     #1802.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 1802.
 6177                                   ;+
 6178                                   ; Verify read/write capability to indirect address field of CSR.
 6179                                   ; Use byte accesses.
 6180                                   ;-
 6181 032032   012702   000017                 MOV     #17,R2          ;SET LOOP COUNT.
 6182 032036   013704   002244                 MOV     CSRA,R4         ;GET CSR ADDRESS.
 6183 032042   110214                 2$:     MOVB    R2,(R4)         ;WRITE COUNT TO CSR.
 6184 032044   111401                         MOVB    (R4),R1         ;READ BACK THE CONTENTS OF THE CSR
 6185 032046   042701   177760                 BIC     #177760,R1      ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
 6186 032052   020102                         CMP     R1,R2           ;CHECK FOR CORRECT DATA WRITTEN/READ.
 6187 032054   001406                        BEQ     4$              ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
 6188                                      ;Report "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
 6189 032056   012737   016712   003774         MOV     #ER1601,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
 6190 032064   005003                         CLR     R3              ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
 6191 032066   005005                         CLR     R5              ;CAUSE REPORT OF LINE 0.
 6192 032070                                   ERROR                   ;  >>>>> ERROR # 1802 <<<<<
      032070   104460                                                                  TRAP    C$ERROR
 6193 032072   005302                 4$:      DEC     R2              ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
 6194 032074   002362                        BGE     2$              ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
 6195                                   ;+
 6196                                   ; Write and verify 16 data patterns in all used bits of all lower bytes of all
 6197                                   ;   registers on all active lines.  Use READ/WRITE accesses.  Before writing
 6198                                   ;   each pattern, clear all the used bits of all active registers.
 6199                                   ;    REGTST routine reports errors with numbers  >>>>> ERROR 1803 - 1805 <<<<<.
 6200                                   ;-
 6201 032076   005237   003770                 INC     ERRNBR          ;SET THE ERROR NUMBER TO 1803.
 6202 032102   012703   177777                 MOV     #-1,R3          ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
 6203 032106   012704   000002                 MOV     #2,R4           ;INDICATE R/W ACCESS, CLEAR FIRST.
```

```
 6204 032112  004737  021220              JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6205                                    ;+
 6206                                    ; Write and verify 16 data patterns in all used bits of all high bytes of all
 6207                                    ;   registers on all active lines.  Use READ/WRITE accesses.  Before writing
 6208                                    ;   each pattern, clear all the used bits of all active registers.
 6209                                    ;    REGTST routine reports errors with numbers  >>>>> ERROR 1806 - 1808 <<<<<
 6210                                    ;-
 6211 032116  012737  003416  003770          MOV     #1806.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
 6212 032124  005403                         NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
 6213 032126  004737  021220                 JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6214                                    ;+
 6215                                    ; Write and verify 16 data patterns in all used bits of all lower bytes of all
 6216                                    ;   registers on all active lines.  Use READ/WRITE accesses.  Before writing
 6217                                    ;   each pattern, set all the used bits of all active registers.
 6218                                    ;    REGTST routine reports errors with numbers  >>>>> ERROR 1809 - 1811 <<<<<.
 6219                                    ;-
 6220 032132  012737  003421  003770          MOV     #1809.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
 6221 032140  005403                         NEG     R3              ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
 6222 032142  005404                         NEG     R4              ;INDICATE R/W ACCESS, SET FIRST.
 6223 032144  004737  021220                 JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6224                                    ;+
 6225                                    ; Write and verify 16 data patterns in all used bits of all high bytes of all
 6226                                    ;   registers on all active lines.  Use READ/WRITE accesses.  Before writing
 6227                                    ;   each pattern, set all the used bits of all active registers.
 6228                                    ;    REGTST routine reports errors with numbers  >>>>> ERROR 1812 - 1814 <<<<<.
 6229                                    ;-
 6230 032150  012737  003424  003770          MOV     #1812.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
 6231 032156  005403                         NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
 6232 032160  004737  021220                 JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6233                                    ;+
 6234                                    ; Print error summary reports if necessary.
 6235                                    ;   The following routine reports errors with number >>>>> ERROR # 1815 <<<<<
 6236                                    ;-
 6237 032164  012737  003427  003770          MOV     #1815.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
 6238 032172  004737  021430                 JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
 6239 032176  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 6240 032202                                 ENDTST
      032202                                                                     L10046:
      032202  104401                                                                 TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - REGBRM -
                                ;++ ******************************************************************************
                                ;*              - Device Register Byte Access Read/Modify/Write Test -
                                ;*
                                ;* This test verifies that the device registers can be read and written
                                ;* correctly using byte accesses in Read/Modify/Write mode.
                                ;*
                                ;-- ******************************************************************************

032204                                  BGNTST
032204                                                                          T18::
                 000022                  TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
032204  012737  000022  002326           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (19)
032212  012737  177777  002352           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
032220  012737  000001  003766           MOV     #1,ERRTYP      ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
032226  012737  003555  003770           MOV     #1901.,ERRNBR  ;SET UP ERROR NUMBER IN THE ERROR TABLE.
032234  012737  012636  003772           MOV     #EM1901,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
032242  005037  002462                   CLR     ERSMRF         ;CLEAR THE ERROR SUMMARY FLAGS.
032246  012700  002464                   MOV     #ERCNTB,R0
032252  004737  020200                   JSR     PC,CLR16W      ;CLEAR THE ERROR COUNTER TABLE.
                                ;+
                                ; Reset the DUT to a known state, do not remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports errors >>>>> 1901 <<<<<.
                                ;-
032256  004737  021456                   JSR     PC,RESETT      ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
032262  103402                           BCS     .+6            ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
032264  000137  032446                   JMP     60$            ;YES,  EXIT THE TEST.
032270  012737  003556  003770           MOV     #1902.,ERRNBR  ;SET THE ERROR REPORT NUMBER TO 1902.
                                ;+
                                ; Verify read/write capability to indirect address field of CSR.
                                ; Use byte accesses.
                                ;-
032276  012702  000017                   MOV     #17,R2         ;SET LOOP COUNT.
032302  013704  002244                   MOV     CSRA,R4        ;GET CSR ADDRESS.
032306  142714  000017           2$:     BICB    #17,(R4)       ;CLEAR THE DUT CSR USING READ/MODIFY/WRITE.
032312  150214                           BISB    R2,(R4)        ;WRITE COUNT TO CSR USING READ/MODIFY/WRITE.
032314  111401                           MOVB    (R4),R1        ;READ BACK THE CONTENTS OF THE CSR
032316  042701  177760                   BIC     #177760,R1     ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
032322  020102                           CMP     R1,R2          ;CHECK FOR CORRECT DATA WRITTEN/READ.
032324  001406                           BEQ     4$             ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                    ;Report "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
032326  012737  016712  003774           MOV     #ER1601,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
032334  005003                           CLR     R3             ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
032336  005005                           CLR     R5             ;CAUSE REPORT OF LINE 0.
032340                                   ERROR                  ;  >>>>> ERROR # 1902 <<<<<
032340  104460                                                                   TRAP    C$ERROR
032342  005302                   4$:     DEC     R2             ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
032344  002360                           BGE     2$             ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
                                ;+
                                ; Write and verify 16 data patterns in all used bits of all lower bytes of all
                                ;   registers on all active lines.  Use READ/MODIFY/WRITE accesses.  Before
                                ;   writing each pattern, clear all the used bits of all active registers.
                                ;    REGTST routine reports errors with numbers  >>>>> ERROR 1903 - 1905 <<<<<.
                                ;-
032346  005237  003770                   INC     ERRNBR         ;SET THE ERROR NUMBER TO 1903.
032352  012703  177777                   MOV     #-1,R3         ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
```

```
 6297 032356  012704  000001            MOV     #1,R4           ;INDICATE R/M/W ACCESS, CLEAR FIRST.
 6298 032362  004737  021220            JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6299                                ;+
 6300                                ; Write and verify 16 data patterns in all used bits of all high bytes of all
 6301                                ;   registers on all active lines.  Use READ/MODIFY/WRITE accesses.  Before
 6302                                ;   writing each pattern, clear all the used bits of all active registers.
 6303                                ;    REGTST routine reports errors with numbers  >>>>> ERROR 1906 - 1908 <<<<<.
 6304                                ;-
 6305 032366  012737  003[illegible]2  003770  MOV     #1906.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
 6306 032374  005403                NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
 6307 032376  004737  021220            JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6308                                ;+
 6309                                ; Write and verify 16 data patterns in all used bits of all lower bytes of all
 6310                                ;   registers on all active lines.  Use READ/MODIFY/WRITE accesses.  Before
 6311                                ;   writing each pattern, set all the used bits of all active registers.
 6312                                ;    REGTST routine reports errors with numbers  >>>>> ERROR 1909 - 1911 <<<<<.
 6313                                ;-
 6314 032402  012737  003565  003770  MOV     #1909.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
 6315 032410  005403                NEG     R3              ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
 6316 032412  005404                NEG     R4              ;INDICATE R/M/W ACCESS, SET FIRST.
 6317 032414  004737  021220            JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6318                                ;+
 6319                                ; Write and verify 16 data patterns in all used bits of all high bytes of all
 6320                                ;   registers on all active lines.  Use READ/MODIFY/WRITE accesses.  Before
 6321                                ;   writing each pattern, set all the used bits of all active registers.
 6322                                ;    REGTST routine reports errors with numbers  >>>>> ERROR 1912 - 1914 <<<<<.
 6323                                ;-
 6324 032420  012737  003570  003770  MOV     #1912.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
 6325 032426  005403                NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
 6326 032430  004737  021220            JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6327                                ;+
 6328                                ; Print error summary reports if necessary.
 6329                                ;   The following routine reports errors with number >>>>> ERROR # 1915 <<<<<
 6330                                ;-
 6331 032434  012737  003573  003770  MOV     #1915.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
 6332 032442  004737  021430            JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
 6333 032446  005037  002352      60$:  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 6334 032452                        ENDTST
      032452                                                                  L10047:
      032452  104401                                                              TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - IDBIT -
                                ;++ ******************************************************************************
                                ;*              - Device Register ID Bit Test -
                                ;*
                                ;*      This test verifies that the DUT STAT register ID bit reads as clear.
                                ;*
                                ;-- ******************************************************************************

                                        BGNTST
                                                                                        T19::
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 032454  012737  000023  002326          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (20)
 032462  012737  177777  002352          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 032470  012737  000001  003766          MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
 032476  012737  003721  003770          MOV     #2001.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
 032504  012737  012713  003772          MOV     #EM2001,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
                                ;+
                                ; Reset the DUT to a known state, do not remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports errors >>>>> 2001 <<<<<.
                                ;-
 032512  004737  021456                 JSR     PC,RESETT       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
 032516  103016                        BCC     60$             ;FATAL RESET ERROR?  YES,  EXIT THE TEST.
                                ;+
                                ; Read the STAT register ID bit and verify that it is clear.
                                ;-
 032520  017701  147526                  MOV     @STATA,R1       ;READ THE STAT REGISTER CONTENTS.
 032524  032701  000400                 BIT     #BIT8,R1        ;CHECK THE ID BIT.
 032530  001411                        BEQ     60$             ;ID BIT CLEAR?  YES, EXIT THE TEST.
 032532  012737  003722  003770          MOV     #2002.,ERRNBR   ;NO,  SET THE ERROR REPORT NUMBER TO 2002.
 032540  012701  012755                 MOV     #EM2002,R1      ;GET THE PROPER ERROR MESSAGE.
 032544  012737  016470  003774          MOV     #ER0503,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
 032552                                 ERROR                   ;ERROR NUMBER >>>>> 2002 <<<<<
 032552  104460                                                                  TRAP    C$ERROR
 032554  005037  002352          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 032560                                 ENDTST
 032560                                                                  L10050:
 032560  104401                                                                  TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE TEST           - NOTXDV -
                                        ;++ ******************************************************************************
                                        ;*                      - No TX_DATA_VALID/No TX_ACTION Test -
                                        ;*      This test verifies that if a data word is written without the
                                        ;*      TX_DATA_VALID bit set, no TX_ACTION will be generated.
                                        ;*      To ensure data is not accidentally transmitted, the test is performed
                                        ;*      in internal loopback, and on all active lines.
                                        ;*
                                        ;-- ******************************************************************************
                                              BGNTST
                                                                                        T20::
                                              TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
032562  012737  000024  002326            MOV     #TNUM,TSTNUM      ;SET UP THE TEST NUMBER.        (21)
032570  012737  177777  002352            MOV     #-1,CTRLCF        ;INDICATE THAT WE ARE IN A TEST.
032576  012737  000001  003766            MOV     #1,ERRTYP         ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
032604  012737  004065  003770            MOV     #2101.,ERRNBR     ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
032612  012737  013030  003772            MOV     #EM2101,ERRMSG    ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
032620  012737  017422  003774            MOV     #ER9101,ERRBLK    ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                        ;+
                                        ; Reset the DUT to a known state, remove the status codes from the fifo.
                                        ;  Clear TX and RX interrupt enable bits in the CSR.
                                        ; This subroutine reports error >>>>> 2101 <<<<<.
                                        ;-
032626  004737  020156                      JSR     PC,CLNRST         ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
032632  103054                            BCC     60$               ;RESET FAILURE?, ABORT THIS TEST.
032634  005237  003770                      INC     ERRNBR            ;SET THE ERROR NUMBER TO 2102.
                                        ;+
                                        ; Set internal loopback on all active lines.
                                        ; Set LPR on all lines to 38.4k baud, 8 bits per character, odd parity,
                                        ; 2 stop bits.
                                        ; Disable transmitters on all active lines.
                                        ;-
032640  013705  002236                      MOV     ACTLNS,R5         ;PASS THE ACTIVE LINE BIT MAP.
032644  012700  000200                      MOV     @200,R0           ;PASS THE LNCTRL CONTENTS.
032650  004737  023516                    JSR     PC,WTWLNC         ;INITIALISE THE LNCTRL REGISTERS.
032654  012700  177670                      MOV     @177670,R0        ;PASS THE LPR CONTENTS.
032660  004737  023572                    JSR     PC,WTWLPR         ;INITIALSE THE LPR REGISTERS ON ALL LINES.
032664  012704  000012                      MOV     #10.,R4           ;PASS DELAY TIME OF 10 MILLI SECS.
032670  004737  020300                    JSR     PC,DELAY          ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
032674  004737  022570                    JSR     PC,TXDSBL         ;DISABLE TRANSMITTERS ON ALL ACTIVE LINES.
                                        ;+
                                        ; Test all active lines individually.
                                        ; Write a data word to the TXCHAR register with TX_DATA_VALID clear.
                                        ; Verify no TX_ACTION is generated.
                                        ;-
032700  013705  002236                      MOV     ACTLNS,R5         ;GET THE ACTIVE LINE BIT MAP.
032704  005004                            CLR     R4                ;CLEAR THE LINE NUMBER COUNTER.
032706  000241                    2$:     CLC                       ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
032710  006005                            ROR     R5                ;SHIFT THE BIT MAP INTO THE CARRY BIT.
032712  103020                          BCC     4$                ;DO NOT TEST THE LINE IF IT IS INACTIVE.
                                        ;+
                                        ; Select the line under test.
                                        ;  Write data word (ASCII <LF>) to TXCHR register with the most significant
                                        ;  bit (TX_DATA_VALID) clear.
                                        ;-
032714  010477  147324                      MOV     R4,@CSRA          ;SELECT THE LINE CURRENTLY UNDER TEST.
032720  012777  000012  147320            MOV     #12,@TXCHA        ;WRITE THE DATA WORD TO THE DUT'S TXCHAR REG.
```

```
                                        ;+
                                        ; Wait for a TX_ACTION to be returned, report error if TX_ACTION found
                                        ; before time-out occurs.
                                        ;-
032726  012701  170002                   MOV     #170002,R1      ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
032732  013702  002244                   MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
032736  004737  023246                  JSR     PC,WAIBIS       ;WAIT FOR TX_ACTION TO COME BACK.
032742  103004                          BCC     4$              ;SKIP ERROR REPORT IF TX-ACTION NOT FOUND.

032744  010401                           MOV     R4,R1           ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
032746  012702  013073                   MOV     #EM2102,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
                                                                ;"TX_ACT FOUND AFTER INVALID DATA WORD WRITTEN"
032752                                  ERROR                   ;               >>>>> ERROR #4102 <<<<<.
032752  104460                                                                          TRAP    C$ERROR
                                ;+
                                ; Verify all active lines have been tested.
                                ;-
032754  005204                  4$:      INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
032756  005705                           TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
032760  001352                          BNE     2$              ;YES; BRANCH TO TEST THE NEXT LINE.
032762  000400                          BR      60$             ;NO; EXIT THIS TEST.

032764  005037  002352          60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
032770                                  ENDTST
032770                                                                          L10051:
032770  10440:                                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - TXDVAL -
                                ;** ****************************************************************************
                                ;*                          - TX_DATA_VALID/TX_ACTION Test -
                                ;*      This test verifies that if a data word is written to the TXCHAR register
                                ;*      with the TX_DATA_VALID bit set, a corresponding TX_ACTION will be
                                ;*      generated.
                                ;*      To ensure data is not accidentally transmitted, the test is performed
                                ;*      in internal loopback, and on all active lines.
                                ;*
                                ;-- ****************************************************************************

032772                                  BGNTST
032772                                                                          T21::
000025                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
032772  012737  000025  002326          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (22)
033000  012737  177777  002352          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
033006  012737  000001  003766          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
033014  012737  004231  003770          MOV     #2201.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
033022  012737  013167  003772          MOV     #EM2201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
033030  012737  017422  003774          MOV     #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;*
                                ; Reset the DUT to a known state, remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports error >>>>> 2201 <<<<<.
                                ;
033036  004737  020156                  JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
033042  103066                          BCC     60$             ;RESET FAILURE?, ABORT THIS TEST.
                                ;*
                                ; Set internal loopback on all active lines.
                                ; Set LPR on all lines to 38.4k baud, 8 bits per character, odd parity,
                                ; 2 stop bits.
                                ; Disable transmitters on all active lines.
                                ;-
033044  013705  002236                  MOV     ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
033050  012700  000200                  MOV     #200,R0         ;PASS THE LNCTRL CONTENTS.
033054  004737  023516                  JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
033060  012700  177670                  MOV     #177670,R0      ;PASS THE LPR CONTENTS.
033064  004737  023572                  JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
033070  012704  000012                  MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI-SECONDS.
033074  004737  020300                  JSR     PC,DELAY        ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
033100  004737  022570                  JSR     PC,TXDSBL       ;DISABLE TRANSMITTERS ON ALL ACTIVE LINES.
                                ;*
                                ; Test all active lines individually.
                                ; Write a data word to the TXCHAR register with TX_DATA_VALID set.
                                ; Verify that a corresponding TX_ACTION is generated.
                                ;
033104  013705  002236                  MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
033110  005004                          CLR     R4              ;CLEAR THE LINE NUMBER COUNTER.
033112  012737  004232  003770  2$:     MOV     #2202.,ERRNBR   ;SET THE ERROR NUMBER TO 2202.
033120  000241                          CLC                     ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
033122  006005                          ROR     R5              ;SHIFT THE BIT MAP INTO THE CARRY BIT.
033124  103032                          BCC     8$              ;DO NOT TEST THE LINE IF IT IS INACTIVE.
                                    ;*
                                    ; Select the line under test.
                                    ;  Write data word (ASCII <LF>) to TXCHR register with the most significant
                                    ;  bit (TX_DATA_VALID) set.
                                    ;-
```

HARDWARE TEST - TXDVAL- SEQ 0168

```
 6507 033126  010477  147112                    MOV     R4,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
 6508 033132  012777  100012  147106            MOV     #100012,@TXCHA  ;WRITE THE DATA WORD TO THE DUT'S TXCHAR REG.
 6509                                          ;+
 6510                                          ; Wait for a TX_ACTION to be returned, report error if no TX_ACTION
 6511                                          ; found before time-out occurs.
 6512                                          ;-
 6513 033140  012701  170002                    MOV     #170002,R1      ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
 6514 033144  013702  002244                    MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 6515 033150  004737  023246                   JSR     PC,WAIBIS       ;WAIT FOR TX_ACTION TO COME BACK.
 6516 033154  103403                           BCS     4$              ;SKIP ERROR REPORT IF TX-ACTION FOUND.
 6517 033156  012702  013224                    MOV     #EM2202,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
 6518                                                                  ;"NO TX_ACT FOUND AFTER VALID DATA WORD TX'D".
 6519 033162  000411                           BR      6$              ;GO REPORT THE ERROR.
 6520                                          ;+
 6521                                          ; Verify TX_ACTION returned from correct line.
 6522                                          ;-
 6523 033164  005237  003770          4$:       INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 2103.
 6524 033170  000302                            SWAB    R2              ;GET THE LINE NUMBER IN THE LOW BYTE.
 6525 033172  042702  177760                    BIC     #177760,R2      ;CLEAR THE UNWANTED BITS.
 6526 033176  020204                            CMP     R2,R4           ;IS IT THE CORRECT LINE NUMBER?.
 6527 033200  001404                           BEQ     8$              ;YES; SKIP THE ERROR REPORT.
 6528 033202  012702  013316                    MOV     #EM2203,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
 6529                                                                  ; "INCORRECT LINE # RETURNED WITH TX_ACT"
 6530 033206  010401                  6$:       MOV     R4,R1           ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
 6531 033210                                    ERROR                   ;                >>>>> ERROR <<<<<.
      033210  104460                                                                              TRAP    C$ERROR
 6532
 6533                                          ;+
 6534                                          ; Verify all active lines have been tested.
 6535                                          ;-
 6536 033212  005204                  8$:       INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
 6537 033214  005705                            TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
 6538 033216  001335                           BNE     2$              ;YES; BRANCH TO TEST THE NEXT LINE.
 6539
 6540 033220  005037  002352          60$:      CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 6541 033224                                    ENDTST
      033224                                                                          L10052:
      033224  104401                                                                          TRAP    C$ETST
```

HARDWARE TEST - TXENBI-

```
                                .SBTTL  HARDWARE TEST           - TXENBI-
                                ;++ ************************************************************************
                                ;*                      - TX_ENABLE (Inactive) Test -
                                ;*      This test verifies that when the line under test's TX_ENABLE bit is
                                ;*      clear, transmission will not take place on that line.
                                ;*      This test is performed in internal loopback, and on all active lines.
                                ;*
                                ;-- ************************************************************************

033226                                  BGNTST
033226                                                                          T22::
        000026                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
033226  012737  000026  002326           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (23)
033234  012737  177777  002352           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
033242  012737  000001  003766           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
033250  012737  004375  003770           MOV     #2301.,ERRNBR  ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
033256  012737  013423  003772           MOV     #EM2301,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
033264  012737  017422  003774           MOV     #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;+
                                ; Reset the DUT to a known state, remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports error >>>>> 2301 <<<<<.
                                ;-
033272  004737  020156                   JSR     PC,CLNRST      ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
033276  103110                           BCC     60$            ;RESET FAILURE?, ABORT THIS TEST.
                                ;+
                                ; Set internal loopback on all active lines.
                                ; Set LPR on all lines to 38.4k baud, 8 bits per character, odd parity,
                                ; 2 stop bits.
                                ; Enable transmitters on all lines.
                                ;-
033300  013705  002236                   MOV     ACTLNS,R5      ;PASS THE ACTIVE LINE BIT MAP.
033304  012700  000200                   MOV     #200,R0        ;PASS THE LNCTRL CONTENTS.
033310  004737  023516                   JSR     PC,WTWLNC      ;INITIALISE THE LNCTRL REGISTERS.
033314  012700  177670                   MOV     #177670,R0     ;PASS THE LPR CONTENTS.
033320  004737  023572                   JSR     PC,WTWLPR      ;INITIALSE THE LPR REGISTERS ON ALL LINES.
033324  012704  000012                   MOV     #10.,R4        ;PASS DELAY TIME OF 10 MILLI-SECONDS.
033330  004737  020300                   JSR     PC,DELAY       ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
033334  012705  000377                   MOV     #MAPLNS,R5     ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
033340  004737  022664                   JSR     PC,TXENBL      ;ENABLE TRANSMITTERS ON ALL LINES.
                                ;+
                                ; Test all active lines individually.
                                ; Disable transmission on each active line.
                                ;-
033344  012703  000001                   MOV     #1,R3          ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
033350  005004                           CLR     R4             ;CLEAR THE LINE NUMBER COUNTER.
033352  012737  004376  003770  20$:      MOV     #2302.,ERRNBR  ;SET THE ERROR NUMBER TO 2302.
033360  030337  002236                   BIT     R3,ACTLNS      ;CHECK IF THE LINE IS ACTIVE
033364  001447                           BEQ     60$            ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
                                ;+
                                ; Clear the TX_ENABLE bit in TBUFFAD2 register.
                                ; Select the line under test.
                                ;   Verify it is clear, report error if set.
                                ;-
033366  010305                           MOV     R3,R5          ;PASS THE BIT MAP OF THE LINE UNDER TEST
033370  004737  022[illegible]           JSR     PC,TXDSBL      ;DISABLE TRANSMISSION ON THE LINE UNDER TEST.
033374  010477  146644                   MOV     R4,@CSRA       ;SELECT THE LINE CURRENTLY UNDER TEST.
```

```
 6599 033400  005777  146654          TST     @TXAD2A         ;VERIFY THE TX_ENABLE BIT IS SET.
 6600 033404  100433                  BMI     4$              ;GO REPORT ERROR IF TX_ENABLE BIT SET.
 6601                                ;+
 6602                                ; Write data word (ASCII <LF>) to TXCHR register.
 6603                                ; Wait for a TX_ACTION to be returned, report error if a TX_ACTION
 6604                                ; is found before time-out occurs.
 6605                                ;-
 6606 033406  012737  004377  003770  MOV     #2303.,ERRNBR   ;SET ERROR NUMBER TO 2303.
 6607 033414  012777  100012  146624  MOV     #100012,@TXCHA  ;WRITE THE DATA WORD TO THE DUT'S TXCHAR REG.
 6608 033422  012701  170002          MOV     #170002,R1      ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
 6609 033426  013702  002244          MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 6610 033432  004737  023246          JSR     PC,WAIBIS       ;WAIT FOR TX_ACTION TO COME BACK.
 6611 033436  103016                  BCC     4$              ;GO REPORT ERROR IF NO TX-ACTION FOUND.
 6612                                ;+
 6613                                ; Wait for the data to appear in the fifo, report error if data found.
 6614                                ;-
 6615 033440  005237  003770          INC     ERRNBR          ;SET ERROR NUMBER TO 2304.
 6616 033444  012701  070012          MOV     #70012,R1       ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
 6617 033450  013702  002244          MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 6618 033454  004737  023246          JSR     PC,WAIBIS       ;WAIT FOR RX_DATA_AVAILABLE TO SET.
 6619 033460  103405                  BCS     4$              ;REPORT ERROR IF DATA RECEIVED IN THE FIFO.
 6620 033462  005237  003770          INC     ERRNBR          ;SET ERROR NUMBER TO 2305.
 6621 033466  017702  146554          MOV     @RBUFA,R2       ;READ THE DATA FROM THE FIFO.
 6622 033472  100004                  BPL     6$              ;SKIP ERROR REPORT IF DATA IS THERE.
 6623
 6624 033474  010401          4$:     MOV     R4,R1           ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
 6625 033476  012702  013461          MOV     #EM2302,R2      ;PASS THE MESSAGE TO BE REPORTED.
 6626                                                        ;"TX_ENABLE BIT BAD ON LINE: nn".
 6627 033502                          ERROR                   ;                >>>>> ERROR <<<<<.
      033502  104460                                                              TRAP    C$ERROR
 6628                                ;+
 6629                                ; Verify all active lines have been tested.
 6630                                ;-
 6631 033504  000241          6$:     CLC                     ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
 6632 033506  006103                  ROL     R3              ;SHIFT THE BIT MAP FOR THE NEXT LINE.
 6633 033510  005204                  INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
 6634 033512  020427  000010          CMP     R4,#NUMLNS      ;HAVE ALL THE LINES BEEN TESTED?.
 6635 033516  002715                  BLT     2$              ;NO; BRANCH TO TEST THE NEXT LINE.
 6636
 6637 033520  005037  002352  60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 6638 033524                          ENDTST
      033524                                                              L10053:
      033524  104401                                                        TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - TXENBA-
                                ;** ************************************************************************
                                ;*                      - TX_ENABLE (Active) Test -
                                ;*      This test verifies that when the TX_ENABLE bit is set in the approriate
                                ;*      line register, transmission will take place on that line.
                                ;*      This test is performed in internal loopback, and on all active lines.
                                ;*
                                ;-- ************************************************************************

                                        BGNTST
033526                                                                          T23::
                        000027           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
033526  012737  000027  002326           MOV     @TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (24)
033534  012737  177777  002352           MOV     @-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
033542  012737  000001  003766           MOV     @1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
033550  012737  004541  003770           MOV     @2401.,ERRNBR  ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
033556  012737  013517  003772           MOV     @EM2401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
033564  012737  017422  003774           MOV     @ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;+
                                ; Reset the DUT to a known state, remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports error >>>>> 2401 <<<<<.
                                ;-
033572  004737  020156                   JSR     PC,CLNRST      ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
033576  103127                          BCC     60$            ;RESET FAILURE?, ABORT THIS TEST.
                                ;+
                                ; Set internal loopback on all active lines.
                                ; Set LPR on all lines to 38.4k baud, 8 bits per character, odd parity,
                                ; 2 stop bits.
                                ; Disable transmitters on all lines.
                                ;-
033600  013705  002236                    MOV     ACTLNS,R5      ;PASS THE ACTIVE LINE BIT MAP.
033604  012700  000200                    MOV     @200,R0        ;PASS THE LNCTRL CONTENTS.
033610  004737  023516                   JSR     PC,WTWLNC      ;INITIALISE THE LNCTRL REGISTERS.
033614  012700  177670                    MOV     @177670,R0     ;PASS THE LPR CONTENTS.
033620  004737  023572                   JSR     PC,WTWLPR      ;INITIALSE THE LPR REGISTERS ON ALL LINES.
033624  012704  000012                    MOV     @10.,R4        ;PASS DELAY TIME OF 10 MILLI-SECONDS.
033630  004737  020300                   JSR     PC,DELAY       ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
033634  012705  000377                    MOV     @MAPLNS,R5     ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
033640  004737  022570                   JSR     PC,TXDSBL      ;DISABLE TRANSMITTERS ON ALL LINES.
                                ;+
                                ; Test all active lines individually.
                                ; Enable transmission on each active line.
                                ;-
033644  012703  000001                    MOV     @1,R3          ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
033650  005004                           CLR     R4             ;CLEAR THE LINE NUMBER COUNTER.
033652  012737  004542  003770  2$:      MOV     @2402.,ERRNBR  ;SET THE ERROR NUMBER TO 2402.
033660  030337  002236                   BIT     R3,ACTLNS      ;CHECK IF THE LINE IS ACTIVE.
033664  001463                          BEQ     8$             ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
                                ;+
                                ; Select the line under test.
                                ;  Set the TX_ENABLE bit in TBUFFAD2 register.
                                ;   Verify it is set, report error if clear.
                                ;-
033666  010305                           MOV     R3,R5          ;PASS THE BIT MAP OF THE LINE UNDER TEST.
033670  004737  022664                   JSR     PC,TXENBL      ;ENABLE TRANSMISSION ON THE LINE UNDER TEST.
033674  012705  000012                    MOV     @10.,R5        ;SET TXCHAR/LOOP COUNT TO 10.
```

```
 6696 033700  010477  146340                    MOV     R4,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
 6697 033704  005777  146350                    TST     @TXAD2A         ;VERIFY THE TX_ENABLE BIT IS SET.
 6698 033710  100045                           BPL     6$              ;GO REPORT ERROR IF TX_ENABLE BIT CLEAR.
 6699                                          ;+
 6700                                          ; Write data word (ASCII <LF>) to TXCHR register.
 6701                                          ; Wait for a TX_ACTION to be returned, report error if no TX_ACTION
 6702                                          ; found before time-out occurs.
 6703                                          ;-
 6704 033712  012737  004543  003770  4$:      MOV     #2403.,ERRNBR   ;SET ERROR NUMBER TO 2403.
 6705 033720  012777  100012  146320           MOV     #100012,@TXCHA  ;WRITE THE DATA WORD TO THE DUT'S TXCHAR REG.
 6706 033726  012701  170002                    MOV     #170002,R1      ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
 6707 033732  013702  002244                    MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 6708 033736  004737  023246                   JSR     PC,WAIBIS       ;WAIT FOR TX_ACTION TO COME BACK.
 6709 033742  103030                           BCC     6$              ;GO REPORT ERROR IF NO TX-ACTION FOUND.
 6710                                          ;+
 6711                                          ; Wait for the data to appear in the fifo, report error if time-out.
 6712                                          ;-
 6713 033744  005237  003770                    INC     ERRNBR          ;SET ERROR NUMBER TO 2404.
 6714 033750  012701  070012                    MOV     #70012,R1       ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
 6715 033754  013702  002244                    MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 6716 033760  004737  023246                   JSR     PC,WAIBIS       ;WAIT FOR RX_DATA_AVAILABLE TO SET.
 6717 033764  103017                           BCC     6$              ;REPORT ERROR IF NO DATA RECEIVED IN THE FIFO.
 6718 033766  005237  003770                    INC     ERRNBR          ;SET ERROR NUMBER TO 2405.
 6719 033772  017702  146250                    MOV     @RBUFA,R2       ;READ THE DATA FROM THE FIFO.
 6720 033776  100012                           BPL     6$              ;GO REPORT ERROR IF THER IS'NT ANY DATA THERE.
 6721 034000  005237  003770                    INC     ERRNBR          ;SET ERROR NUMBER TO 2406.
 6722 034004  000302                           SWAB    R2              ;PUT THE LINE NUMBER IN THE LOW BYTE.
 6723 034006  042702  177760                    BIC     #177760,R2      ;CLEAR THE UNWANTED BITS.
 6724 034012  020204                           CMP     R2,R4           ;DID THE DATA COME FROM THE CORRECT LINE?.
 6725 034014  001003                           BNE     6$              ;NO; GO REPORT THE ERROR.
 6726 034016  005305                           DEC     R5              ;DECREMENT THE TXCHAR/LOOP COUNTER.
 6727 034020  001334                           BNE     4$              ;LOOP TO TX THE NEXT CHAR.
 6728 034022  000404                           BR      8$              ;GO TEST THE NEXT LINE.
 6729
 6730 034024  010401                  6$:      MOV     R4,R1           ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
 6731 034026  012702  013461                    MOV     #EM2302,R2      ;PASS THE MESSAGE TO BE REPORTED.
 6732                                          ;"TX_ENABLE BIT BAD ON LINE: nn".
 6733 034032                                   ERROR                   ;               >>>>> ERROR <<<<<.
      034032  104460                                                                    TRAP    C$ERROR
 6734                                          ;+
 6735                                          ; Verify all active lines have been tested.
 6736                                          ;-
 6737 034034  010305                  8$:      MOV     R3,R5           ;PASS THE BIT MAP OF THE LINE UNDER TEST.
 6738 034036  004737  022570                   JSR     PC,TXDSBL       ;CLEAR THE TX ENABLE BIT ON THIS LINE.
 6739 034042  000241                           CLC                     ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
 6740 034044  006103                           ROL     R3              ;SHIFT THE BIT MAP FOR THE NEXT LINE.
 6741 034046  005204                           INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
 6742 034050  020427  000010                   CMP     R4,#NUMLNS      ;HAVE ALL THE LINES BEEN TESTED?.
 6743 034054  002676                           BLT     2$              ;NO; BRANCH TO TEST THE NEXT LINE.
 6744
 6745 034056  005037  002352          60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 6746 034062                                   ENDTST
      034062                                                                    L10054:
      034062  104401                                                                    TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - INTA -
                                ;++ ******************************************************************************
                                ;*                      - Interrupt Test -
                                ;*      This test verifies that the Device Under Test (DUT) will generate
                                ;*      reception and transmission interrupts correctly.  This test does
                                ;*      not depend on the use of the serial line transmission or reception
                                ;*      capabilities of the DUT.  The lines are put in internal loopback
                                ;*      to minimize any external effects that could be caused on devices
                                ;*      attached to the serial lines.
                                ;*
                                ;-- ******************************************************************************

034064                                  BGNTST
034064                                                                  T24::
                 000030                  TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
034064  012737  000030  002326           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (26)
034072  012737  177777  002352           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
034100  012737  000001  003766           MOV     #1,ERRTYP      ;SET ERROR FATAL ERROR TYPE IN ERROR TABLE.
034106  012737  003101  003770           MOV     #1601.,ERRNBR  ;SET FIRST ERROR REPORT NUMBER IN ERROR TABLE.
034114  012737  013553  003772           MOV     #EM2601.,ERRMSG ;SET TEST ERROR MESSAGE IN ERROR TABLE.
                                ;+
                                ; Reset the DUT to a known state, do not remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports errors from >>>>> 2601 thru 2602 <<<<<.
                                ;-
034122  004737  021456                   JSR     PC,RESETT      ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
034126  103402                          BCS     2$             ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
034130  000137  035070                   JMP     60$            ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
034134  012737  005053  003770   2$:      MOV     #2603.,ERRNBR  ;SET THE ERROR REPORT NUMBER TO 2603.
                                ;+
                                ; Enable transmitters on all lines.
                                ;-
034142  012705  000377           4$:      MOV     #MAPLNS,R5     ;PASS ACTIVE LINE BIT MAP.
034146  004737  022664                   JSR     PC,TXENBL      ;ENABLE TRANSMISSION ON ALL LINES.
                                ;+
                                ; Test reception interrupts.
                                ;    Set up for RX and TX interrupts:
                                ;    RX interrupt service routine inputs a char and counts the interrupt.
                                ;    TX interrupt service routine counts TX interrupts.
                                ;-
034152  005037  002334                   CLR     RXINTC         ;CLEAR THE RX INTERRUPT COUNTER.
034156  005037  002336                   CLR     RXINTF         ;CLEAR THE RX INTERRUPT FLAGS.
034162  005037  002340                   CLR     TXINTC         ;CLEAR THE TX INTERRUPT COUNTER.
034166  012737  002726  002324           MOV     #BUFBAS,BUFPTR ;LOAD THE BUFFER PTR WITH THE BUFFER BASE ADR.
034174  106427  000240                  MTPS    #PRI05         ;DISABLE DEVICE INTERRUPTS.
034200                                  SETVEC  RXVECA,#RXINPT,#PRI05 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
034200  012746  000240                                                  MOV     #PRI05,-(SP)
034204  012746  024036                                                  MOV     #RXINPT,-(SP)
034210  013746  002232                                                  MOV     RXVECA,-(SP)
034214  012746  000003                                                  MOV     #3,-(SP)
034220  104437                                                          TRAP    C$SVEC
034222  062706  000010                                                  ADD     #10,SP
034226                                  SETVEC  TXVECA,#CACHTX,#PRI05 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
034226  012746  000240                                                  MOV     #PRI05,-(SP)
034232  012746  023650                                                  MOV     #CACHTX,-(SP)
034236  013746  002234                                                  MOV     TXVECA,-(SP)
034242  012746  000003                                                  MOV     #3,-(SP)
```

```
                034246  104437                                                          TRAP    C$SVEC
                034250  062706  000010                                                  ADD     #10,SP
 6794           034254  106427  000000                  MTPS    #PRI00          ;ALLOW INTERRUPTS.
 6795                                                   ;+
 6796                                                   ;Enable reception interrupts.
 6797                                                   ;Delay 4 ms to allow time for the interrupts to take place.
 6798                                                   ;Disable reception interrupts.
 6799                                                   ;-
 6800           034260  004737  022236                   JSR     PC,RXIE1        ;ENABLE THE RECEPTION INTERRUPTS.
 6801           034264  012704  000004                   MOV     #4,R4           ;PASS 4 MS COUNT TO THE DELAY ROUTINE.
 6802           034270  004737  020300                   JSR     PC,DELAY        ;DELAY 4 MILLI-SECONDS.
 6803           034274  004737  022204                   JSR     PC,RXIE0        ;DISABLE RECEPTION INTERRUPTS.
 6804                                                   ;+
 6805                                                   ; Verify that the correct interrupts took place.
 6806                                                   ;    Test the int counter to verify that interrupts took place.
 6807                                                   ;-
 6808           034300  005737  002334                   TST     RXINTC          ;CHECK THE RX INTERRUPT COUNT.
 6809           034304  001017                          BNE     6$              ;SKIP THE FOLLOWING ERRORS IF COUNT <> 0.
 6810                                                   ;+
 6811                                                   ; Determine reason for no RX interrupts and print proper error message.
 6812                                                   ;-
 6813           034306  012701  013762                   MOV     #EM2604,R1      ;SET UP MSG IN CASE "RX.DATA.AVAIL IS CLR".
 6814           034312  032777  00020C  145724           BIT     #BIT7,@CSRA     ;TEST THE RX.DATA.AVAIL BIT OF THE CSR.
 6815           034320  001416                          BEQ     8$              ;GO REPORT ERROR IF RX.DATA.AVAIL IS CLR.
 6816           034322  012701  013674                   MOV     #EM2603,R1      ;SET UP MSG IN CASE "DATA.VALID IS CLEAR".
 6817           034326  032777  100000  145712           BIT     #BIT15,@RBUFA   ;TEST THE DATA.VALID BIT OF THE FIFO.
 6818           034334  001410                          BEQ     8$              ;GO REPORT ERROR IF DATA.VALID IS CLEAR.
 6819           034336  012701  013603                   MOV     #EM2602,R1      ;SET UP MSG,"DATA.VALID IS SET".
 6820           034342  000405                          BR      8$              ;GO REPORT THE ERROR.
 6821                                                   ;+
 6822                                                   ; If RX ints occurred with RX.DATA.AVAIL clear, report the error.
 6823                                                   ;-
 6824           034344  005737  002336          6$:      TST     RXINTF          ;CHECK THE RX INTERRUPT FLAGS.
 6825           034350  100006                          BPL     10$             ;SKIP THE ERROR IF FLAG IS CLEAR.
 6826           034352  012701  014056                   MOV     #EM2605,R1      ;SET UP THE PROPER MESSAGE.
 6827                                                   ;+
 6828                                                   ; Report the error wh ch has been found.
 6829                                                   ;-
 6830           034356                          8$:      ERRDF   2603,EM2601,ER0503;             >>>>> ERROR #2603 <<<<<.
                034356  104455                                                                  TRAP    C$ERDF
                034360  005053                                                                  .WORD   2603
                034362  013553                                                                  .WORD   EM2601
                034364  016470                                                                  .WORD   ER0503
 6831                                                   ;+
 6832                                                   ; Verify that no TX interrupts have been generated so far in this test.
 6833                                                   ;-
 6834           034366  013702  002340          10$:     MOV     TXINTC,R2       ;LOAD # OF TX INTERRUPTS FOR ER0504 RTN.
 6835           034372  001406                          BEQ     12$             ;SKIP ERROR IF NO TX INTERRUPTS.
 6836                                                  ;REPORT "TX INTERRUPTS(S) RECEIVED WITH TX INTERRUPTS DISABLED."
 6837           034374  012701  010003                   MOV     #EM0526,R1      ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
 6838           034400                                  ERRDF   2604,EM2601,ER0504;             >>>>> ERROR #2604 <<<<<.
                034400  104455                                                                  TRAP    C$ERDF
                034402  005054                                                                  .WORD   2604
                034404  013553                                                                  .WORD   EM2601
                034406  016514                                                                  .WORD   ER0504
 6839                                                   ;+
 6840                                                   ; Clean out the interrupt vectors used in this test.
```

```
6841
6842 034410  106427  000240          ;-
6843 034414                    12$:    MTPS    #PRIO5          ;DISABLE DEVICE INTERRUPTS.
       034414  013700  002232          CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
       034420  104436                                                  MOV     RXVECA,R0
6844 034422                                                           TRAP    C$CVEC
       034422  013700  002234          CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
       034426  104436                                                  MOV     TXVECA,R0
6845                                                                  TRAP    C$CVEC
6846                                   ;+
6847                                   ; Test transmission interrupts.
6848                                   ;    Set up for RX and TX interrupts:
6849                                   ;    RX interrupt service routine counts RX interrupts.
6850                                   ;    TX interrupt service routine counts the interrupt and sets flags.
6851 034430  005037  002334            ;-
6852 034434  005037  002340             CLR     RXINTC          ;CLEAR THE RX INTERRUPT COUNTER.
6853 034440  005037  002342             CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNTER.
6854 034444                             CLR     TXINTF          ;CLEAR THE RX INTERRUPT FLAGS.
       034444  012746  000240          SETVEC  RXVECA,#CACHRX,#PRIO5 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
       034450  012746  023622                                          MOV     #PRIO5,-(SP)
       034454  013746  002232                                          MOV     #CACHRX,-(SP)
       034460  012746  000003                                          MOV     RXVECA,-(SP)
       034464  104437                                                  MOV     #3,-(SP)
       034466  062706  000010                                          TRAP    C$SVEC
6855 034472                                                           ADD     #10,SP
       034472  012746  000240          SETVEC  TXVECA,#TXINTR,#PRIO5 ;SET UP INT VECTOR TO TX INT ROUTINE.
       034476  012746  024144                                          MOV     #PRIO5,-(SP)
       034502  013746  002234                                          MOV     #TXINTR,-(SP)
       034506  012746  000003                                          MOV     TXVECA,-(SP)
       034512  104437                                                  MOV     #3,-(SP)
       034514  062706  000010                                          TRAP    C$SVEC
6856 034520  106427  000000                                           ADD     #10,SP
6857                                       MTPS    #PRIO0          ;ALLOW INTERRUPTS.
6858                                   ;+
6859                                   ; Verify that the TX_ACTION bit is clear.
6860 034524  012705  000022            ;-
6861 034530  012701  000144                 MOV     #18.,R5         ;INITIALIZE THE LOOP COUNTER.
6862 034534  012702  100000                 MOV     #100.,R1        ;SET 100 MS TIME-OUT.
6863 034540  013704  002244                 MOV     #BIT15,R2       ;SELECT TX_ACTION BIT TO TEST.
6864 034544  012703  100000                 MOV     CSRA,R4         ;PASS DUT CSR AS THE WORD TO TEST.
6865 034550  004737  020454          14$:    MOV     #BIT15,R3       ;WAIT FOR TX_ACTION TO BE SET.
6866 034554  103020                       JSR     PC,MSLOOP       ;WAIT UP TO 100 MS FOR TX_ACTION SET.
6867 034556  005003                       BCC     20$             ;IF TIME-OUT, CONSIDER TX_ACTION CLEAR.
6868 034560  004737  020454                CLR     R3              ;NOW, WAIT FOR TX_ACTION CLEAR.
6869 034564  103005                       JSR     PC,MSLOOP       ;WAIT UP TO 100 MS FOR TX_ACTION CLEAR.
6870 034566  005305                       BCC     16$             ;IF TIME-OUT, REPORT TX_ACTION WON'T CLEAR.
6871 034570  001365                       DEC     R5              ;DECREMENT THE TX_ACTION SET COUNTER.
6872                                      BNE     14$             ;LOOP IF NOT TOO MANY TX_ACTIONS FOUND.
6873 034572  012701  014200           ;REPORT "TX_ACTION SET REPEATEDLY AFTER RESET, NO DATA SENT."
6874 034576  000402                       MOV     #EM2607,R1      ;SELECT ERROR MESSAGE.
6875 034600  012701  014274                BR      18$             ;GO TO REPORT THE ERROR.
6876 034604                         16$:    MOV     #EM2608,R1      ;SELECT TX_ACTION STUCK SET MSG.
       034604  104455                18$:    ERRDF   2605,EM2606,ER0503;             >>>>> ERROR #2605 <<<<<.
       034606  005055                                                                  TRAP    C$ERDF
       034610  014141                                                                  .WORD   2605
       034612  016470                                                                  .WORD   EM2606
6877 034614  000424                                                                   .WORD   ER0503
                                       BR      24$             ;GO TO TEST WITH TX_ACTION SET.
```

```
                                        ;+
                                        ; Verify that no interrupts occur with TX_ACTION clear.
                                        ;-
034616  004737  023012          20$:    JSR     PC,TXIE1        ;ENABLE TX_INTERRUPTS.
034622  012704  000062                  MOV     #50.,R4         ;PASS 50 MS TIME TO THE DELAY ROUTINE.
034626  004737  020300                  JSR     PC,DELAY        ;DELAY 50 MILLI-SECONDS TO ALLOW INTS TO OCCUR.
034632  005737  002340                  TST     TXINTC          ;TEST THE TX INTERRUPT COUNT.
034636  001413                          BEQ     24$             ;SKIP THE ERROR IF NO TX INTERRUPTS.
034640  012701  014200                  MOV     #EM2607,R1      ;SELECT MESSAGE IN CASE TX INT FLAG CLEAR.
034644  005737  002342                  TST     TXINTF          ;TEST THE TX INTERRUPT FLAGS.
034650  100002                          BPL     22$             ;GO REPORT ERROR IF TX FLAG IS CLEAR.
034652  012701  014345                  MOV     #EM2609,R1      ;TX FLAG IS SET, SELECT PROPER ERROR MESSAGE.
                                        ;REPORT "TRANSMIT INTERRUPT TEST ERROR:..."
034656                          22$:    ERRDF   2606,EM2606,ER0503;             >>>>> ERROR #2606 <<<<<.
034656  104455                                                                  TRAP    C$ERDF
034660  005056                                                                  .WORD   2606
034662  014141                                                                  .WORD   EM2606
034664  016470                                                                  .WORD   ER0503
                                        ;+
                                        ; Prepare TX interrupt counter and flags.
                                        ;-
034666  005037  002340          24$:    CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNT.
034672  005037  002342                  CLR     TXINTF          ;CLEAR THE TX INTERRUPT FLAGS.
                                        ;+
                                        ; Set up line parameters for transmission.
                                        ;-
034676  012705  C00377                  MOV     #MAPLNS,R5      ;PASS ACTIVE LINES BIT MAP.
034702  012700  000200                  MOV     #200,R0         ;PASS INERT STATE, INTERNAL LOOPBACK.
034706  004737  023516                  JSR     PC,WTWLNC       ;DISABLE RECPTION AND DMA, ETC. ON DUT.
034712  012700  156430                  MOV     #156430,R0      ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
034716  004737  023572                  JSR     PC,WTWLPR       ;WRITE TO ALL LPR REGISTERS.
                                        ;+
                                        ; Send a null char to each line.
                                        ;-
034722  013701  002246                  MOV     TXCHA,R1        ;SET UP TXCHAR REGISTER ADDRESS.
034726  012702  100000                  MOV     #100000,R2      ;SET CHARACTER TO BE TRANSMITTED = NULL.
034732  004737  023546                  JSR     PC,WTWLNS       ;SEND NULL CHAR TO EACH LINE.
                                        ;+
                                        ; Delay 250 milli-seconds to allow interrupts to occur.
                                        ;-
034736  012704  000372                  MOV     #250.,R4        ;SET UP FOR 250 MS DELAY.
034742  004737  020300                  JSR     PC,DELAY        ;WAIT 250 MS.
                                        ;+
                                        ; Verify that TX interrupts occurred.
                                        ;-
034746  005737  002340                  TST     TXINTC          ;CHECK THE TX INTERRUPT COUNTER.
034752  001007                          BNE     26$             ;SKIP THE FOLLOWING ERROR IF WE GOT TX INTS.
                                        ;+
                                        ; Determine the reason that we received no interrupts.
                                        ;-
034754  012701  014424                  MOV     #EM2610,R1      ;SET UP MSG IN CASE "TX_ACTION IS SET".
034760  005777  145260                  TST     @CSRA           ;CHECK THE DUT CSR.
034764  100407                          BMI     28$             ;GO TO REPORT ERROR IF TX_ACTION IS SET.
034766  012701  014516                  MOV     #EM2611,R1      ;SET UP "TX_ACTION NOT SET" MESSAGE.
                                        ;+
                                        ; Check to verify that TX_ACTION was set for each interrupt.
                                        ;-
```

HARDWARE TEST - INTA -

```
6931 034772  005737  002342          26$:     TST     TXINTF          ;CHECK THE TX INTERRUPT FLAGS.
6932 034776  100006                           BPL     30$             ;SKIP ERROR IF TX_ACTION CLR FLAG IS CLEAR.
6933 035000  012701  014345                    MOV     #EM2609,R1      ;SET UP TX INT WITH "TX_ACTION CLR" MSG.
6934                                       ;+
6935                                       ; Report "TRANSMIT INTERRUPT TEST ERROR:...."
6936                                       ;-
6937 035004                                28$:    ERRDF   2607,EM2606,ER0503;             >>>>> ERROR #2607 <<<<<.
     035004  104455                                                                              TRAP    C$ERDF
     035006  005057                                                                              .WORD   2607
     035010  014141                                                                              .WORD   EM2606
     035012  016470                                                                              .WORD   ER0503
6938                                       ;+
6939                                       ; Verify that no TX interrupts have been generated so far in this test.
6940                                       ;-
6941 035014  013702  002334          30$:     MOV     RXINTC,R2       ;LOAD # OF RX INTERRUPTS FOR ER0504 RTN.
6942 035020  001406                           BEQ     32$             ;SKIP ERROR IF NO RX INTERRUPTS.
6943 035022  012701  007713                    MOV     #EM0525,R1      ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
6944                                       ;REPORT "RX INTERRUPTS(S) RECEIVED WITH RX INTERRUPTS DISABLED."
6945 035026                                    ERRDF   2608,EM2606,ER0504;             >>>>> ERROR #2608 <<<<<.
     035026  104455                                                                              TRAP    C$ERDF
     035030  005060                                                                              .WORD   2608
     035032  014141                                                                              .WORD   EM2606
     035034  016514                                                                              .WORD   ER0504
6946                                       ;+
6947                                       ; Disable interrupts and clean out the interrupt vectors used in this test.
6948                                       ;-
6949 035036  005001                ;32$:     CLR     R1              ;CLEAR BOTH TRANSMITTER
6950 035040  004737  022760                    JSR     PC,TXIE0        ; INTERRUPT ENABLE AND RECEIVER
6951 035044  004737  022204                    JSR     PC,RXIE0        ; INTERRUPT ENABLE BITS IN THE DUT CSR.
6952 035050  106427  000240                    MTPS    #PRI05          ;DISABLE DEVICE INTERRUPTS.
6953 035054                                    CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
     035054  013700  002232                                                                      MOV     RXVECA,R0
     035060  104436                                                                              TRAP    C$CVEC
6954 035062                                    CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
     035062  013700  002234                                                                      MOV     TXVECA,R0
     035066  104436                                                                              TRAP    C$CVEC
6955
6956 035070  005037  002352          60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6957
6958 035074                                    ENDTST
     035074                                                                          L10055:
     035074  104401                                                                              TRAP    C$ETST
6959
```

```
                                .SBTTL  HARDWARE TEST           - BRLEVA -
                                ;++ ****************************************************************************
                                ;*                      - BR Level Test B -
                                ;*      This test verifies that the Device Under Test (DUT) will generate
                                ;*      reception and transmission interrupts at the correct BR level.
                                ;*      This test does not depend on the use of the serial line transmission
                                ;*      or reception capabilities of the DUT.  The lines are put in internal
                                ;*      loopback to minimize any external effects that could be caused on
                                ;*      devices attached to the serial lines.
                                ;*
                                ;-- ****************************************************************************

035076                                  BGNTST
035076                                                                          T25::
        000031                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
035076  012737  000031  002326   MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (30)
035104  012737  177777  002352   MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
035112  012737  000001  003766   MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
035120  012737  005671  003770   MOV    #3001.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
035126  012737  014601  003772   MOV    #EM3001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
035134  005037  002462           CLR    ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
                                ;+
                                ; Reset the DUT to a known state, do not remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports errors from >>>>> 3001 thru 3002 <<<<<.
                                ;-
035140  004737  021456          JSR     PC,RESETT       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
035144  103402                  BCS     2$              ;SKIP AROUND ABORTING TEST IF NO ERR R FOUND.
035146  000137  035736          JMP     60$             ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
035152  012737  005673  003770  2$:     MOV    #3003.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 3003.
                                ;+
                                ; Enable transmitters on all lines.
                                ;-
035160  012705  000377          4$:     MOV    #MAPLNS,R5      ;PASS ACTIVE LINE BIT MAP.
035164  004737  022664          JSR     PC,TXENBL       ;ENABLE TRANSMISSION ON ALL LINES.
                                ;+
                                ; Generate a transmission interrupt request.
                                ; Processor priority should be at 7 disabling ints.
                                ;-
035170  106427  000340          MTPS    #PRI07          ;DISABLE ALL INTERRUPTS.
035174                          SETVEC  TXVECA,#TXINTR,#PRI07 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
035174  012746  000340                                                  MOV     #PRI07,-(SP)
035200  012746  024144                                                  MOV     #TXINTR,-(SP)
035204  013746  002234                                                  MOV     TXVECA,-(SP)
035210  012746  000003                                                  MOV     #3,-(SP)
035214  104437                                                          TRAP    C$SVEC
035216  062706  000010                                                  ADD     #10,SP
                                ;+
                                ; Set up DUT for transmission interrupts:
                                ;  Set up internal loopback.
                                ;  Set up line parameters for transmission.
                                ;-
                                 MOV    #MAPLNS,R5      ;PASS ACTIVE LINES BIT MASK.
                                 MOV    #200,R0         ;PASS INERT STATE, INTERNAL LOOPBACK.
                                JSR     PC,WTWLNC       ;DISABLE RECPTION AND DMA, ETC. ON DUT.
                                 MOV    #156430,R0      ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
                                JSR     PC,WTWLPR       ;WRITE INTO ALL LPR REGISTERS.
```

```
                                        ;+
                                        ; Send a null char to each line.
                                        ;-
035246  013701  002246                      MOV     TXCHA,R1        ;SET UP TXCHAR REGISTER ADDRESS.
035252  012702  100000                      MOV     #100000,R2      ;SET CHARACTER TO BE TRANSMITTED = NULL.
035256  004737  023546                     JSR     PC,@TWLNS       ;SEND NULL CHAR TO EACH LINE.
                                        ;+
                                        ; Delay 50 ms to allow time for the interrupt to be generated.
                                        ;-
035262  012704  000062                      MOV     #50.,R4         ;PASS 50 MS TIME TO THE DELAY ROUTINE.
035266  004737  020300                     JSR     PC,DELAY        ;DELAY 50 MILLI-SECONDS.
                                    ;+
                                    ; Generate a reception interrupt request.
                                    ;-
035272                                      SETVEC  RXVECA,#RXBRRT,#PRIO7 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
035272  012746  000340                                                                  MOV     #PRIO7,-(SP)
035276  012746  023746                                                                  MOV     #RXBRRT,-(SP)
035302  013746  002232                                                                  MOV     RXVECA,-(SP)
035306  012746  000003                                                                  MOV     #3,-(SP)
035312  104437                                                                          TRAP    C$SVEC
035314  062706  000010                                                                  ADD     #10,SP
                                    ;+
                                    ; Set up for the loop which tests the interrupt BR levels.
                                    ;-
035320  012705  000340                      MOV     #340,R5         ;SET UP THE PRIORITY LEVEL TO 7.
035324  005003                             CLR     R3              ;CLEAR THE RX PRIORITY STORE AND FLAGS.
035326  005002                             CLR     R2              ;CLEAR THE TX PRIORITY STORE AND FLAGS.
                                        ;+
                                        ; Enable TX and RX interrupts.
                                        ; Processor priority should be at 7 disabling the interrupts.
                                        ;-
035330  004737  022236                      JSR     PC,RXIE1        ;ENABLE RECEIVER INTERRUPTS.
035334  004737  023012                     JSR     PC,TXIE1        ;ENABLE TRANSMITTER INTERRUPTS.
                                    ;+
                                    ; Loop, lowering the processor priority until the DUT interrupts on RX and TX.
                                    ;-
035340  005037  002340              6$:     CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNTER.
035344  005037  002342                     CLR     TXINTF          ;CLEAR THE TX INTERRUPT FLAGS.
035350  005037  002334                     CLR     RXINTC          ;CLEAR THE RX INTERRUPT COUNTER.
035354  005037  002336                     CLR     RXINTF          ;CLEAR THE RX INTERRUPT FLAGS.
035360  106405                           MTPS    R5              ;SET PROCESSOR PRIORITY TO THE SELECTED VALUE.
035362  012704  000001                      MOV     #1,R4           ;PASS 1 MS COUNT TO THE DELAY ROUTINE.
035366  004737  020300                     JSR     PC,DELAY        ;DELAY 1 MS TO ALLOW INTERRUPTS TO OCCUR.
                                        ;+
                                        ; Determine if any RX DUT interrup s occurred.
                                        ; Log the processor priority for the RX interrupt if first RX int.
                                        ;-
035372  005737  002334                      TST     RXINTC          ;CHECK THE RECEIVE INTERRUPT COUNTER.
035376  001412                           BEQ     8$              ;SKIP THE PRIORITY LOG IF NO RX INT OCCURRED.
                                        ;+
                                        ; If this is the first RX interrupt, log the priority.
                                        ;-
035400  005703                             TST     R3              ;CHECK THE RX PRIORITY STORE AND FLAGS.
035402  001010                           BNE     8$              ;GOTO TEST FOR TX INTS IF NOT THE FIRST RX INT.
035404  010503                              MOV     R5,R3           ;LOG THE PRESENT PRIORITY IN THE RX PRIO STORE.
035406  052703  100000                      BIS     #BIT15,R3       ;SET THE RX INT HAS OCCURRED FLAG.
035412  013700  002336                      MOV     RXINTF,R0       ;GET THE RX INTERRUPT ROUTINE FLAGS.
```

```
 7062 035416  042700  137777              BIC     #137777,R0      ;CLEAR ALL BUT THE TX INT ERROR FLAG.
 7063 035422  050003                      BIS     R0,R3           ;IF TX INT ERROR, SET BIT 14 OF THE PRIO FLAGS.
 7064                                  ;+
 7065                                  ; Determine if any TX DUT interrupts have occurred.
 7066                                  ; Log the present processor priority if this is the first TX interrupt.
 7067                                  ;-
 7068 035424  005737  002340          8$:     TST     TXINTC          ;CHECK THE TRANSMIT INTERRUPT COUNTER.
 7069 035430  001405                      BEQ     10$             ;SKIP THE PRIORITY LOG IF NO TX INT OCCURRED.
 7070                                      ;+
 7071                                      ; If this is the first TX interrupt, log the priority.
 7072                                      ;-
 7073 035432  005702                       TST     R2              ;CHECK THE TX PRIORITY STORE AND FLAGS.
 7074 035434  100403                      BMI     10$             ;SKIP THE LOGGING IF NOT FIRST TX INTERRUPT.
 7075 035436  010502                       MOV     R5,R2           ;LOG THE PRESENT PRIORITY IN THE TX PRIO STORE.
 7076 035440  052702  100000               BIS     #BIT15,R2       ;SET THE TX INT HAS OCCURRED FLAG.
 7077                                  ;+
 7078                                  ; Select next processor priority.
 7079                                  ; Test for both RX and TX interrupts having occurred, loop if not.
 7080                                  ;-
 7081 035444  162705  000040          10$:     SUB     #40,R5          ;DECREMENT PRIORITY LEVEL BY ONE.
 7082 035450  002402                      BLT     12$             ;GOTO CHECK FOR ERRORS IF BELOW PRIORITY ZERO.
 7083 035452  030203                       BIT     R2,R3           ;AND PRIO FLAGS TOGETHER, ALTER NONE OF THEM.
 7084 035454  100331                      BPL     6$              ;LOOP IF RX AND TX INTS HAVEN'T BOTH OCCURRED.
 7085                                  ;+
 7086                                  ; Disable interrupts and clear interrupt vectors.
 7087                                  ;-
 7088 035456  106427  000340          12$:    MTPS    #PRI07          ;DISABLE ALL INTERRUPTS.
 7089 035462                               CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
      035462  013700  002232                                                  MOV     RXVECA,R0
      035466  104436                                                          TRAP    C$CVEC
 7090 035470                               CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
      035470  013700  002234                                                  MOV     TXVECA,R0
      035474  104436                                                          TRAP    C$CVEC
 7091                                  ;+
 7092                                  ; Verify that RX and TX interrupts occurred,
 7093                                  ; at the proper BR level, and
 7094                                  ; in the proper order.
 7095                                  ;    Determine if TX interrupt occurred.
 7096                                  ;-
 7097 035476  005702                       TST     R2              ;DETERMINE WHETHER TX INT OCCURRED OR NOT.
 7098 035500  100414                      BMI     16$             ;SKIP THESE ERRORS IF TX INT OCCURRED.
 7099                                  ;+
 7100                                  ; Determine reason that no TX int occurred.
 7101                                  ;-
 7102 035502  012701  014424               MOV     #EM2610,R1      ;SELECT "NO TX INT FROM TX.ACTION" MESSAGE.
 7103 035506  005777  144532               TST     @CSRA           ;CHECK THE TX.ACTION BIT OF THE DUT CSR.
 7104 035512  100402                      BMI     14$             ;SKIP TX.ACTION CLR MSG SELECTION IF IT IS SET.
 7105 035514  012701  014516               MOV     #EM2611,R1      ;SELECT "TX.ACTION CLEAR AFTER CHARS SENT" MSG.
 7106                                  ;REPORT "INTERRUPT BR LEVEL TEST ERROR:"
 7107 035520                          14$:    ERRDF   3003,EM3001,ER0503;             >>>>> ERROR #3003 <<<<<.
      035520  104455                                                          TRAP    C$ERDF
      035522  005673                                                          .WORD   3003
      035524  014601                                                          .WORD   EM3001
      035526  016470                                                          .WORD   ER0503
 7108 035530  000423                      BR      18$             ;SKIP THE BR LEVEL CHECK, NO TX INT OCCURRED.
 7109                                  ;+
 7110                                  ; Verify that the TX interrupt was at the proper BR level.
```

```
                                        ;-
035532  010204                  16$:     MOV     R2,R4           ;CALCULATE THE BR LEVEL
035534  042704  177400                   BIC     #177400,R4      ; THAT THE TRANMSIT
035540  006204                           ASR     R4              ; INTERRUPT WAS
035542  006204                           ASR     R4              ; REQUESTED AT, WHICH
035544  006204                           ASR     R4              ; IS ONE GREATER THAN
035546  006204                           ASR     R4              ; THE PROCESSOR PRIORITY
035550  006204                           ASR     R4              ; LEVEL AT WHICH THE
035552  005204                           INC     R4              ; TRANSMIT INTERRUPT OCCURRED.
035554  113705  002242                   MOVB    BRLEVL,R5       ;GET THE EXPECTED INTERRUPT BR LEVEL.
035560  120405                           CMPB    R4,R5           ;COMARE THE INTERRUPT BR LEVEL WITH EXPECTED.
035562  001406                         BEQ     18$             ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
                                     ;REPORT "TX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
035564  012701  014726                   MOV     #EM3003,R1      ;SELECT THE ERROR MESSAGE FOR THE ERROR CALL.
035570                                 ERRDF   3004,EM3001,ER3001;             >>>>> ERROR #3004 <<<<<.
035570  104455                                                                          TRAP    C$ERDF
035572  005674                                                                          .WORD   3004
035574  014601                                                                          .WORD   EM3001
035576  017070                                                                          .WORD   ER3001
                                        ;+
                                        ; Determine if RX interrupt occurred.
                                        ;-
035600  005703                  18$:     TST     R3              ;CHECK THE RX INT OCCURRED FLAG.
035602  100415                         BMI     22$             ;SKIP THESE ERRORS IF RX INT OCCURRED.
                                        ;+
                                        ; Determine reason that no RX int occurred.
                                        ;-
035604  012701  013603                   MOV     #EM2602,R1      ;SELECT "NO RX INT FROM TX.ACTION" MSG.
035610  032777  000200  144426           BIT     #BIT7,@CSRA     ;CHECK THE RX.DATA.AVAIL BIT OF THE DUT CSR.
035616  001002                         BNE     20$             ;SKIP RX.DATA.AVAIL CLR MSG IF BIT IS SET.
035620  012701  014632                   MOV     #EM3002,R1      ;SELECT "NO RX.DATA.AVAIL AFTER RESET" MSG.
                                     ;REPORT "INTERRUPT BR LEVEL TEST ERROR:"
035624                           20$:    ERRDF   3005,EM3001,ER0503;             >>>>> ERROR #3005 <<<<<.
035624  104455                                                                          TRAP    C$ERDF
035626  005675                                                                          .WORD   3005
035630  014601                                                                          .WORD   EM3001
035632  016470                                                                          .WORD   ER0503
035634  000423                         BR      24$             ;SKIP THE BR CHECK IF NO RX INT OCCURRED.
                                        ;+
                                        ; Verify that the RX interrupt was at the proper BR level.
                                        ;-
035636  010304                  22$:     MOV     R3,R4           ;CALCULATE THE BR LEVEL
035640  042704  177400                   BIC     #177400,R4      ; THAT THE RECEIVE
035644  006204                           ASR     R4              ; INTERRUPT WAS
035646  006204                           ASR     R4              ; REQUESTED AT, WHICH
035650  006204                           ASR     R4              ; IS ONE GREATER THAN
035652  006204                           ASR     R4              ; THE PROCESSOR PRIORITY
035654  006204                           ASR     R4              ; LEVEL AT WHICH THE
035656  005204                           INC     R4              ; RECEIVE INTERRUPT OCCURRED.
035660  113705  002242                   MOVB    BRLEVL,R5       ;GET THE EXPECTED INTERRUPT BR LEVEL.
035664  120405                           CMPB    R4,R5           ;COMARE THE INTERRUPT BR LEVEL WITH EXPECTED.
035666  001406                         BEQ     24$             ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
                                     ;REPORT "RX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
035670  012701  015002                   MOV     #EM3004,R1      ;SELECT ERROR MESSAGE FOR THE ERROR CALL.
035674                                 ERRDF   3006,EM3001,ER3001;             >>>>> ERROR #3006 <<<<<.
035674  104455                                                                          TRAP    C$ERDF
035676  005676                                                                          .WORD   3006
```

```
035700  014601                                                          .WORD   EM3001
035702  017070                                                          .WORD   ER3001
7158
7159                                    ;+
7160                                    ; Test for interrupts occuring in the proper order.
7161                                    ;-
7162 0357(-  032703  040000     24$:    BIT     #BIT14,R3       ;CHECK THE IMPROPER INT ORDER ERROR FLAG.
7163 035710  001406                    BEQ     26$             ;SKIP ERROR REPORT IF ERROR DID NOT OCCUR.
7164                                ;REPORT "TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS RX INT."
7165 035712  012701  015056            MOV     #EM3005,R1      ;SELECT THE ERROR MESSAGE FOR INDIRECT PRINT.
7166 035716                            ERRDF   3007,EM3001,ER0503;             >>>>> ERROR #3007 <<<<<.
035716  104455                                                          TRAP    C$ERDF
035720  005677                                                          .WORD   3007
035722  014601                                                          .WORD   EM3001
035724  016470                                                          .WORD   ER0503
7167                                    ;+
7168                                    ; Clean up, exit the test.
7169                                    ;-
7170 035726  004737  022760     26$:    JSR     PC,TXIE0        ;CLEAR TRANSMITTER INTERRUPTS.
7171 035732  004737  022204             JSR     PC,RXIE0        ;CLEAR RECEIVER INTERRUPTS.
7172 035736  005037  002352     60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7173 035742  106427  000240            MTPS    #PRI05          ;DISABLE ALL BUT LTC INTERRUPTS.
7174 035746                            ENDTST
035746                                                                  L10056:
035746  104401                                                                  TRAP    C$ETST
7175
```

```
                                .SBTTL  HARDWARE TEST           - DIABMP -
                                ;++ ************************************************************************
                                ;*                          - Diagnostic field (BMP) Test -
                                ;*      This test verifies that a request to the DUT to report BMP status
                                ;*       odes is complied with, within the specified time.
                                ;*      All active lines are tested.
                                ;-- ************************************************************************

035750                                  BGNTST
035750                                                                          T26::
        000032                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
035750  012737  000032  002326           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (31)
035756  012737  177777  002352           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
035764  012737  000001  003766           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
035772  012737  006035  003770           MOV    #3101.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
036000  012737  015150  003772           MOV    #EM3101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
036006  012737  017422  003774           MOV    #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;+
                                ; Reset the DUT to a known state, remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports error >>>>> 3101 <<<<<.
                                ;-
036014  004737  020156                  JSR    PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
036020  103077                          BCC    60$             ;RESET FAILURE?, ABORT THIS TEST.
                                ;+
                                ; Test all active lines individually.
                                ; Write the request code to the diagnostic field in the LPR register.
                                ; Verify that a BMP code is returned within the correct time.
                                ;-
036022  013705  002236                   MOV    ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
036026  005004                           CLR    R4              ;CLEAR THE LINE NUMBER COUNTER.
036030  013703  002244                   MOV    CSRA,R3         ;GFT THE ADDRESS OF THE DUT'S CSR.
036034  000241                  2$:      CLC                    ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
036036  006005                           ROR    R5              ;SHIFT THE BIT MAP INTO THE CARRY BIT.
036040  103064                          BCC    8$              ;DO NOT TEST THE LINE IF IT IS INACTIVE.
                                ;+
                                ; Select the line under test.
                                ;  Write the BMP request code to the DIAG field in the LPR register.
                                ;-
036042  012737  006036  003770           MOV    #3102.,ERRNBR   ;SET THE ERROR NUMBER TO 3102.
036050  010413                           MOV    R4,(R3)         ;SELECT THE LINE CURRENTLY UNDER TEST.
036052  052777  000002  144170           BIS    #2,@LPRA        ;WRITE THE BMP REQUEST CODE TO THE LPR.
                                ;+
                                ; Wait for BMP request code to be cleared, report error if time-out
                                ; occurs.
                                ;-
036060  012701  010764                   MOV    #10764,R1       ;TEST BIT 1, TIMEOUT OF 500 MILLI SECS.
036064  013702  002250                   MOV    LPRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
036070  004737  023172                  JSR    PC,WAIBIC       ;WAIT FOR REQUEST CODE TO CLEAR.
036074  103042                          BCC    6$              ;GO REPORT ERROR IF CODE DID NOT CLEAR IN TIME.
                                ;+
                                ; Wait for BMP code to appear in the fifo, report error if time-out
                                ; occurs.
                                ;-
036076  005237  003770                   INC    ERRNBR          ;SET ERROR NUMBER TO 3103.
036102  012701  070012                   MOV    #70012,R1       ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
036106  013702  002244                   MOV    CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
```

```
 7233 036112  004737  023246                JSR     PC,WAIBIS       ;WAIT FOR RX_DATA_AVAILABLE TO SET.
 7234 036116  103031                        BCC     6$              ;GO REPORT ERROR IF CODE DID NOT CLEAR IN TIME.
 7235                                      ;+
 7236                                      ; Read the BMP code (if it is there) from the RBUF register.
 7237                                      ; Determine if it is a valid BMP code,
 7238                                      ; Verify the BMP code was received from the correct channel.
 7239                                      ; If the BMP code does not indicate DUT running ok, then save it on
 7240                                      ; the queue to be reported in a later test.
 7241                                      ;-
 7242 036120  005237  003770                INC     ERRNBR          ;SET ERROR NUMBER TO 3104.
 7243 036124  017702  144116                MOV     @RBUFA,R2       ;GET THE BMP CODE FROM THE FIFO.
 7244 036130  100024                       BPL     6$              ;GO REPORT ERROR IF NO BMP CODE FOUND.
 7245 036132  005237  003770                INC     ERRNBR          ;SET ERROR NUMBER TO 3105.
 7246 036136  012700  170301                MOV     #170301,R0      ;SET-UP A BMP CODE MASK.
 7247 036142  040200                        BIC     R2,R0           ;TRY TO CLEAR THE BMP MASK.
 7248 036144  001016                       BNE     6$              ;GO REPORT ERROR IF IT IS NOT A VALID BMP CODE.
 7249 036146  005237  003770                INC     ERRNBR          ;SET THE ERROR NUMBER TO 3106.
 7250 036152  010200                        MOV     R2,R0           ;COPY THE BMP CODE.
 7251 036154  000300                        SWAB    R0              ;PUT THE LINE NUMBER IN THE LOW BYTE.
 7252 036156  042700  177760                BIC     #177760,R0      ;CLEAR THE UNWANTED BITS.
 7253 036162  120400                        CMPB    R4,R0           ;DID THE BMP CODE COME FROM THE CORRECT LINE?.
 7254 036164  001006                       BNE     6$              ;NO; GO REPORT ERROR.
 7255 036166  120227  000305                CMPB    R2,#305         ;IS THE BMP CODE A "GOOD ONE"?.
 7256 036172  001407                       BEQ     8$              ;YES; SKIP SAVING THE BMP CODE ON THE QUEUE.
 7257 036174  004737  022262                JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
 7258 036200  000404                       BR      8$              ;GO SEE IF THERE ARE ANY MORE LINE TO TEST.
 7259
 7260 036202  010401              6$:       MOV     R4,R1           ;PASS THE LINE NUMBER TO BE REPORTED.
 7261 036204  012702  015204                MOV     #EM3102,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
 7262                                                               ;"BMP REQUEST BIT BAD ON LINE:"
 7263 036210                               ERROR                   ;               >>>>> ERROR <<<<<.
      036210  104460                                                                 TRAP    C$ERROR
 7264                                    ;+
 7265                                    ; Verify all active lines have been tested.
 7266                                    ;-
 7267 036212  005204              8$:       INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
 7268 036214  005705                        TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
 7269 036216  001306                       BNE     2$              ;YES; BRANCH TO TEST THE NEXT LINE.
 7270 036220  005037  002352      60$:      CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7271 036224                               ENDTST
      036224                                                                 L10057:
      036224  104401                                                         TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - REPBMP -
                                ;++ ******************************************************************************
                                ;*                      - Report any BMP codes in the queue -
                                ;*      This is a pseudo-test used to report any BMP codes that were found
                                ;*      in the DUT's FIFO during previous test, and logged in the BMP code
                                ;*      queue.
                                ;*      It is unlikely that running this pseudo-test alone will produce any
                                ;*      error reports.
                                ;*
                                ;-- ******************************************************************************
036226                                  BGNTST
036226                                                                          T27::
                        000033           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
036226  012737  000033  002326           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (93)
036234  012737  177777  002352           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
036242  013702  002524                   MOV     BMPCQP,R2      ;GET THE CONTENTS OF THE POINTER.
036246  012703  002526                   MOV     #BMPCQB,R3     ;GET THE START ADDRESS OF THE QUEUE.
036252  020203                           CMP     R2,R3          ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
036254  001411                         BEQ     60$            ;EXIT NO CODES IN THE QUEUE.
                                ;+
                                ; There is at least one BMP code in the queue. Report the error.
                                ;-
                                        ;Report error BMP CODE FOUND IN TEST nn, BMP CODE:nnnnnn"

036256  012701  016013                  MOV     #EM9304,R1     ;PASS THE FIRST MESSAGE TO BE REORTED.
036262                                  ERRDF   9301,EM9301,ER9301 ;            >>>>> ERROR #9301 <<<<<.
036262  104455                                                                  TRAP    C$ERDF
036264  022125                                                                  .WORD   9301
036266  015676                                                                  .WORD   EM9301
036270  017450                                                                  .WORD   ER9301

036272  012737  002526  002524           MOV     #BMPCQB,BMPCQP ;SET POINTER BACK TO THE BEGINING OF THE QUE.

036300  005037  002352          60$:     CLR     CTRLCF         ;INDICATE THAT WE ARE NOT WITHIN A TEST.
036304                                  ENDTST
036304                                                                  L100:
036304  104401                                                          TRAP    C$ETST
```

```
                        ;*****************************************************************************
                        ;
                        ;               VDHA.HWQ
                        ;
                        ;*****************************************************************************


                        .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.
                        ;--

036306                          BGNHRD
036306  000022                                                      .WORD L10061-L$HARD/2
036310                                                          L$HARD::

                        ;DEVICE CSR ADDRESS QUESTION:
036310                          GPRMA   HWPTQ1,0,0,160000,177776,YES
036310  000031                                                      .WORD   T$CODE
036312  036354                                                      .WORD   HWPTQ1
036314  160000                                                      .WORD   T$LOLIM
036316  177776                                                      .WORD   T$HILIM

                        ;DEVICE INTERRUPT VECTOR QUESTION:
036320                          GPRMA   HWPTQ2,2,0,40,776,YES
036320  001031                                                      .WORD   T$CODE
036322  036372                                                      .WORD   HWPTQ2
036324  000040                                                      .WORD   T$LOLIM
036326  000776                                                      .WORD   T$HILIM

                        ;ACTIVE LINES BIT MAP QUESTION:
036330                          GPRMD   HWPTQ3,4,0,MAPLNS,0,177777,YES
036330  002032                                                      .WORD   T$CODE
036332  036425                                                      .WORD   HWPTQ3
036334  000377                                                      .WORD   MAPLNS
036336  000000                                                      .WORD   T$LOLIM
036340  177777                                                      .WORD   T$HILIM

                        ;INTERRUPT BR LEVEL QUESTION:
036342                          GPRMD   HWPTQ5,6,0,377,0,6,YES
036342  003032                                                      .WORD   T$CODE
036344  036453                                                      .WORD   HWPTQ5
036346  000377                                                      .WORD   377
036350  000000                                                      .WORD   T$LOLIM
036352  000006                                                      .WORD   T$HILIM


036354                          ENDHRD
                                                                    .EVEN
036354                                                          L10061:
```

```
7357 036354    103    123    122         HWPTQ1: .ASCIZ  /CSR ADDRESS: /
     036357    040    101    104
     036362    104    122    105
     036365    123    123    072
     036370    040    000
7358 036372    111    116    124         HWPTQ2: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
     036375    105    122    122
     036400    125    120    124
     036403    040    126    105
     036406    103    124    117
     036411    122    040    101
     036414    104    104    122
     036417    105    123    123
     036422    072    040    000
7359 036425    101    103    124         HWPTQ3: .ASCIZ  /ACTIVE LINE BIT MAP: /
     036430    111    126    105
     036433    040    114    111
     036436    116    105    040
     036441    102    111    124
     036444    040    115    101
     036447    120    072    040
     036452    000
7360 036453    111    116    124         HWPTQ5: .ASCIZ  /INTERRUPT BR LEVEL: /
     036456    105    122    122
     036461    125    120    124
     036464    040    102    122
     036467    040    114    105
     036472    126    105    114
     036475    072    040    000
7361
7362                                             .EVEN
```

```
 7365                           ;*****************************************************************************
*
 7366                           ;
 7367                           ;               VDHA.SWQ
 7368                           ;
 7369                           ;*****************************************************************************
*
 7371
 7372
 7373                           .SBTTL  SOFTWARE PARAMETER CODING SECTION
 7374
 7375                           ;++
 7376                           ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
 7377                           ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 7378                           ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 7379                           ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 7380                           ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 7381                           ; WITH THE OPERATOR.
 7382                           ;--
 7383
 7384 036500                           BGNSFT
      036500  000013                                                        .WORD L10062-L$SOFT/2
      036502                                                          L$SOFT::
 7385
 7394                           ;UNIT NUMBER PRINTOUT QUESTION:
 7395 036502                           GPRML   SWPTQ1,0,20,YES
      036502  000130                                                        .WORD   T$CODE
      036504  036530                                                        .WORD   SWPTQ1
      036506  000020                                                        .WORD   20
 7396                           ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
 7397 036510                           GPRMD   SWPTQ2,2,D,177777,0,177777,YES
      036510  001052                                                        .WORD   T$CODE
      036512  036604                                                        .WORD   SWPTQ2
      036514  177777                                                        .WORD   177777
      036516  000000                                                        .WORD   T$LOLIM
      036520  177777                                                        .WORD   T$HILIM
 7398                           ;ROM VERSION PRINTOUT ON FIRST PASS QUESTION:
 7399 036522                           GPRML   SWPTQ3,0,1,YES
      036522  000130                                                        .WORD   T$CODE
      036524  036673                                                        .WORD   SWPTQ3
      036526  000001                                                        .WORD   1
 7400
 7401                                  .EVEN
 7402
 7403 036530                           ENDSFT
      036530                                                              .EVEN
                                                                    L10062:
 7404
 7405
 7412 036530  122 105 120              SWPTQ1: .ASCIZ  /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /
      036533  117 122 124
      036536  040 125 116
      036541  111 124 040
      036544  116 125 115
      036547  102 105 122
      036552  040 101 123
      036555  040 105 101
      036560  103 110 040
      036563  125 116 111
```

```
                036566  124     040     111
                036571  123     040     124
                036574  105     123     124
                036577  105     104     072
                036602  040     000
7413            036604  116     125     115     SWPTQ2: .ASCIZ  /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /
                036607  102     105     122
                036612  040     117     106
                036615  040     111     116
                036620  104     111     126
                036623  111     104     125
                036626  101     114     040
                036631  104     101     124
                036634  101     040     105
                036637  122     122     117
                036642  122     123     040
                036645  124     117     040
                036650  122     105     120
                036653  117     122     124
                036656  040     117     116
                036661  040     101     040
                036664  114     111     116
                036667  105     072     040
                036672  000
7414            036673  122     117     115     SWPTQ3: .ASCIZ  /ROM VERSION PRINTOUT ON THE FIRST PASS: /
                036676  040     126     105
                036701  122     123     111
                036704  117     116     040
                036707  120     122     111
                036712  116     124     117
                036715  125     124     040
                036720  117     116     040
                036723  124     110     105
                036726  040     106     111
                036731  122     123     124
                036734  040     120     101
                036737  123     123     072
                036742  040     000
7415                                    .EVEN
```

SOFTWARE PARAMETER CODING SECTION

```
                                ;*****************************************************************************
                                ;
                                ;               FVTSKL6.P11
                                ;
                                ;*****************************************************************************


036744                          $PATCH::
036744                                  .BLKW   24


037014                                  LASTAD
                                                                                        .EVEN
037014  000000                                                                         .WORD   0
037016  000000                                                                         .WORD   0
037020                          L$LAST::
037020                                  ENDMOD


000001                                  .END
```

SEQ 0191

Symbol table

```
ACTLNS  002236 G        CTRLCF  002352 G        C$RPT = 000025          EM0401  007252 G        EM2608  014274 G
ADR   = 000020 G        C$AU  = 000052          C$SEFG= 000046          EM0402  007321 G        EM2609  014345 G
ADRPTR  020140 G        C$AUTO= 000061          C$SPRI= 000041          EM0501  007471 G        EM2610  014424 G
ALTFLD  017630 G        C$BRK = 000022          C$SVEC= 000037          EM0502  007537 G        EM2611  014516 G
ASSEMB= 000010          C$BSEG= 000004          C$TOME= 000076          EM0525  007713 G        EM3001  014601 G
BCOUNT  002372 G        C$BSUB= 000002          DELAY   020300 G        EM0526  010003 G        EM3002  014632 G
BITTBL  002410 G        C$CLCK= 000062          DFPTBL  002214 G        EM0601  010073 G        EM3003  014726 G
BIT0  = 000001 G        C$CLEA= 000012          DIAGMC= 000000          EM0602  010127 G        EM3004  015002 G
BIT00 = 000001 G        C$CLOS= 000035          DRADRT  002244 G        EM0603  010307 G        EM3005  015056 G
BIT01 = 000002 G        C$CLP1= 000006          DROP    025160          EM0701  010472 G        EM3101  015150 G
BIT02 = 000004 G        C$CPBF= 000074          DR00MG  006004 G        EM0702  010527 G        EM3102  015204 G
BIT03 = 000010 G        C$CPME= 000075          DR02MG  006010 G        EM0703  010707 G        EM9009  015245 G
BIT04 = 000020 G        C$CVEC= 000036          DR04MG  006015 G        EM0801  011072 G        EM9010  015271 G
BIT05 = 000040 G        C$DCLN= 000044          DR06MG  006021 G        EM0802  011132 G        EM9014  015315 G
BIT06 = 000100 G        C$DODU= 000051          DR10MG  006026 G        EM0901  011205 G        EM9017  015411 G
BIT07 = 000200 G        C$DRPT= 000024          DR12MG  006035 G        EM0902  011236 G        EM9018  015522 G
BIT08 = 000400 G        C$DU  = 000053          DR14MG  006046 G        EM1001  011272 G        EM9019  015532 G
BIT09 = 001000 G        C$EDIT= 000000          DR16MG  006057 G        EM1002  011336 G        EM9020  015547 G
BIT1  = 000002 G        C$ERDF= 000055          EDROP   025236          EM1003  011423 G        EM9022  015573 G
BIT10 = 002000 G        C$ERHR= 000056          EF.CON= 000036 G        EM1101  011501 G        EM9023  015612 G
BIT11 = 004000 G        C$ERRO= 000060          EF.NEW= 000035 G        EM1201  011525 G        EM9024  015634 G
BIT12 = 010000 G        C$ERSF= 000054          EF.PWR= 000034 G        EM1202  011544 G        EM9026  015652 G
BIT13 = 020000 G        C$ERSO= 000057          EF.RES= 000037 G        EM1203  011630 G        EM9301  015676 G
BIT14 = 040000 G        C$ESCA= 000010          EF.STA= 000040 G        EM1204  011706 G        EM9302  015716 G
BIT15 = 100000 G        C$ESEG= 000005          EF0503  004134 G        EM1205  011742 G        EM9303  015746 G
BIT2  = 000004 G        C$ESUB= 000003          EF0505  004141 G        EM1301  011766 G        EM9304  016013 G
BIT3  = 000010 G        C$ETST= 000001          EF1401  004214 G        EM1302  012012 G        ENDETB  003726 G
BIT4  = 000020 G        C$EXIT= 000032          EF1402  004316 G        EM1401  012051 G        ENDIT   025050
BIT5  = 000040 G        C$FREQ= 000101          EF1601  004353 G        EM1402  012101 G        ERCNTB  002464 G
BIT6  = 000100 G        C$FRME= 000100          EF1602  004377 G        EM1403  012167 G        ERLTBL  002726 G
BIT7  = 000200 G        C$GETB= 000026          EF1603  004441 G        EM1404  012242 G        ERRBLK  003774 G
BIT8  = 000400 G        C$GETW= 000027          EF1604  004503 G        EM1405  012314 G        ERRMSG  003772 G
BIT9  = 001000 G        C$GMAN= 000043          EF3001  004600 G        EM1406  012327 G        ERRNBR  003770 G
BMPCQB  002526 G        C$GPHR= 000042          EF3002  004647 G        EM1407  012342 G        ERRTYP  003766 G
BMPCQE  002726 G        C$GPRI= 000040          EF9001  004716 G        EM1408  012354 G        ERSMRF  002462 G
BMPCQP  002524 G        C$INIT= 000011          EF9002  005000 G        EM1601  012362 G        ER0101  016070 G
BOE   = 000400 G        C$INLP= 000020          EF9003  005057 G        EM1604  012[illegible]5 G        ER0201  016406 G
BRLEVL  002242 G        C$MANI= 000050          EF9004  005113 G        EM1701  012513 G        ER0503  016470 G
BUFBAS  002726 G        C$MAP  = 000102         EF9005  005150 G        EM1801  012570 G        ER0504  016514 G
BUFEND  003726 G        C$MEM  = 000031         EF9006  005201 G        EM1901  012636 G        ER1401  016562 G
BUFMID  003326 G        C$MMU  = 000103         EF9010  005225 G        EM2001  012713 G        ER1601  016712 G
BUFPTR  002324 G        C$MSG  = 000023         EF9016  005324 G        EM2002  012755 G        ER1603  017010 G
BUF3QT  003526 G        C$OPNR= 000034          EF9017  005421 G        EM2101  013030 G        ER3001  017070 G
CACHRX  023622 G        C$OPNW= 000104          EF9018  005475 G        EM2102  013073 G        ER9004  017160 G
CACHTX  023650 G        C$PNTB= 000014          EF9019  005602 G        EM2201  013167 G        ER9007  017262 G
CALMSL  017702 G        C$PNTF= 000017          EF9301  005626 G        EM2202  013224 G        ER9008  017340 G
CKTRAP  020126 G        C$PNTS= 000016          EF9302  005704 G        EM2203  013316 G        ER9101  017422 G
CLKBRL  002356 G        C$PNTX= 000015          EM0101  020534 G        EM2301  013423 G        ER9301  017450 G
CLKCSR  002354 G        C$PUTB= 000072          EM0102  020620 G        EM2302  013461 G        EVL   = 000004 G
CLKHRZ  002362 G        C$PUTW= 000073          EM0103  006067 G        EM2401  013517 G        E$END = 002100
CLKINT  023676 G        C$QIO  = 000377         EM0201  006125 G        EM2601  013553 G        E$LOAD= 000035
CLKVEC  002360 G        C$RDBU= 000007          EM0202  006173 G        EM2602  013603 G        F$AU  = 000015
CLNRST  020156 G        C$REFG= 000047          EM0203  006346 G        EM2603  013674 G        F$AUTO= 000020
CLR16W  020200 G        C$REL  = 000077         EM0204  006511 G        EM2604  013762 G        F$BGN = 000040
CNTERR  020222 G        C$RESE= 000033          EM0301  006670 G        EM2605  014056 G        F$CLEA= 000007
CSRA    002244 G        C$REVI= 000004          EM0302  006733 G        EM2606  014141 G        F$DU  = 000016
CSR0  = 000000 G        C$RFLA= 000021          EM0303  007073 G        EM2607  014200 G        F$END = 000041
```

Symbol table

```
F$HARD= 000004          I$PTAB= 000041          L$PROT  024226 G        L10054  034062          RMATBB  002304 G
F$HW  = 000013          I$PWR = 000041          L$PRT   002112 G        L10055  035074          ROLDAP  021570 G
F$INIT= 000006          I$RPT = 000041          L$REPP  002062 G        L10056  035746          RSTRPT  021622 G
F$JMP = 000050          I$SEG = 000041          L$REV   002010 G        L10057  036224          RXBDTX= 000030 G
F$MOD = 000000          I$SETU= 000041          L$RPT   024220 G        L10060  036304          RXBETX= 000020 G
F$MSG = 000011          I$SFT = 000041          L$SOFT  036502 G        L10061  036354          RXBFUL= 000100 G
F$PROT= 000021          I$SRV = 000041          L$SPC   002056 G        L10062  036530          RXBRRT  023746 G
F$PWR = 000017          I$SUB = 000041          L$SPCP  002020 G        MAPLNS= 000377 G        RXIE0   022204 G
F$RPT = 000012          I$TST = 000041          L$SPTP  002024 G        MFUNIT  004076 G        RXIE1   022236 G
F$SEG = 000003          J$JMP = 000167          L$STA   002030 G        MMENAB  002404 G        RXINPT  024036 G
F$SOFT= 000005          LNCTRA  002254 G        L$SW    002226 G        MMPRES  002402 G        RXINTC  002334 G
F$SRV = 000010          LNCTRO= 000010 G        L$TEST  002114 G        MMSR0   002400 G        RXINTF  002336 G
F$SUB = 000002          LOE   = 040000 G        L$TIML  002014 G        MSG1    016174 G        RXVECA  002232 G
F$SW  = 000014          LOT   = 000010 G        L$UNIT  002012 G        MSG2    016252 G        R0SLOT= 000002 G
F$TEST= 000001          LPCSLT= 000036 G        L10000  002224          MSG3    016331 G        R1SLOT= 000004 G
GETPRM  024634          LPRA    002250 G        L10001  002232          MSLCNT  002376 G        R2SLOT= 000006 G
GPRS0B  002450 G        LPR0  = 000004 G        L10002  016172          MSLGET  020340 G        R3SLOT= 000010 G
G$CNT0= 000200          L$ACP   002110 G        L10003  016466          MSLOOP  020454 G        R4SLOT= 000012 G
G$DELM= 000372          L$APT   002036 G        L10004  016512          MSTICK  002374 G        R5SLOT= 000014 G
G$DISP= 000003          L$AU    025244 G        L10005  016560          NDERPT  002230 G        SAVBMP  022262 G
G$EXCP= 000400          L$AUT   002070 G        L10006  016710          NEWPAS  024614          SFPTBL  002226 G
G$HILI= 000002          L$AUTO  025076 G        L10007  017006          NEWRES  024606          SKPSTS  022330 G
G$LOLI= 000001          L$CCP   002106 G        L10010  017066          NEWSTA  024300          STATA   002252 G
G$NO  = 000000          L$CLEA  025100 G        L10011  017156          NUMLNS= 000010 G        STAT0 = 000006 G
G$OFFS= 000400          L$CO    002032 G        L10012  017260          OOPS    020470 G        SVCGBL= 000000
G$OFSI= 000376          L$DEP0  002011 G        L10013  017336          OPTION  002226 G        SVCINS= 000001
G$PRMA= 000001          L$DESC  004046 G        L10014  017420          O$APTS= 000000          SVCSUB= 000001
G$PRMD= 000002          L$DESP  002076 G        L10015  017446          O$AU  = 000000          SVCTAG= 000001
G$PRML= 000000          L$DEVP  002060 G        L10016  017626          O$BGNR= 000001          SVCTST= 000001
G$RADA= 000140          L$DISP  002124 G        L10017  024224          O$BGNS= 000001          SWAP0   022406 G
G$RADB= 000000          L$DLY   002116 G        L10021  025074          O$DU  = 000001          SWPTQ1  036530
G$RADD= 000040          L$DTP   002040 G        L10022  025076          O$ERRT= 000001          SWPTQ2  036604
G$RADL= 000120          L$DTYP  002034 G        L10023  025132          O$GNSW= 000001          SWPTQ3  036673
G$RADO= 000020          L$DU    025134 G        L10024  025242          O$POIN= 000001          S$LSYM= 010000
G$XFER= 000004          L$DUT   002072 G        L10025  025250          O$SETU= 000000          TIMER1  002364 G
G$YES = 000010          L$DVTY  004036 G        L10026  025540          PAR0A   002406 G        TIMER2  002366 G
HELP  = 000000          L$EF    002052 G        L10027  025754          PASCNT  002332 G        TIMER3  002370 G
HOE   = 100000 G        L$ENVI  002044 G        L10030  026204          PCSLOT= 000016 G        TNUM  = 000033 G
HWPTQ1  036354          L$ERRT  003766 G        L10031  026366          PNT   = 001000 G        TP4FLG  002346 G
HWPTQ2  036372          L$ETP   002102 G        L10032  026544          PREGRT  004020 G        TP4RTN  024122 G
HWPTQ3  036425          L$EXP1  002046 G        L10033  026746          PREG05  003776          TP4VEC  002344 G
HWPTQ5  036453          L$EXP4  002064 G        L10034  027140          PRI   = 002000 G        TSABRT  022456 G
IBE   = 010000 G        L$EXP5  002066 G        L10035  027332          PRI00 = 000000 G        TSTNUM  002326 G
IDU   = 000040 G        L$HARD  036310 G        L10036  027504          PRI01 = 000040 G        TXAD1A  002256 G
IER   = 020000 G        L$HIME  002120 G        L10037  027704          PRI02 = 000100 G        TXAD10= 002012 G
IESTAT  002330 G        L$HPCP  002016 G        L10040  030116          PRI03 = 000140 G        TXAD2A  002260 G
ISR   = 000100 G        L$HPTP  002022 G        L10041  030364          PRI04 = 000200 G        TXAD20= 000014 G
IXE   = 004000 G        L$HW    002214 G        L10042  030726          PRI05 = 000240 G        TXBFCA  002262 G
I$AU  = 000041          L$ICP   002104 G        L10043  031312          PRI06 = 000300 G        TXBFC0= 000016 G
I$AUTO= 000041          L$INIT  024234 G        L10044  031522          PRI07 = 000340 G        TXCHA   002246 G
I$CLN = 000041          L$LADP  002026 G        L10045  031736          PUFIF0  020716 G        TXCHR0= 000002 G
I$DU  = 000041          L$LAST  037020 G        L10046  032202          RBUFA   002246 G        TXDSBL  022570 G
I$HRD = 000041          L$LOAD  002100 G        L10047  032452          RBUF0 = 000002 G        TXENBL  022664 G
I$INIT= 000041          L$LUN   002074 G        L10050  032560          RDPDR   021000 G        TXIE0   022760 G
I$MOD = 000041          L$MREV  002050 G        L10051  032770          REGTST  021220 G        TXIE1   023012 G
I$MSG = 000041          L$NAME  002000 G        L10052  033224          REPSMR  021430 G        TXINTC  002340 G
I$PRGT= 000040          L$PRIO  002042 G        L10053  033524          RESETT  021456 G        TXINTF  002342 G
```

Symbol table

```
TXINTR  024144 G        T$SAVL= 177777          T$$PRO= 010020          T2      025542 G        UAM   = 000200 G
TXVECA  002234 G        T$SEGL= 177777          T$$RPT= 010017          T20     032562 G        UNBTTB  002264 G
T$ARGC= 000003          T$SUBN= 000000          T$$SOF= 010062          T21     032772 G        UNITN   002240 G
T$CODE= 000130          T$TAGL= 177777          T$$SW = 010001          T22     033226 G        UNSDIV  023036 G
T$ERRN= 022125          T$TAGN= 010063          T$$TES= 010060          T23     033526 G        WAIBIC  023172 G
T$EXCP= 000000          T$TEMP= 000000          T1      025252 G        T24     034064 G        WAIBIS  023246 G
T$FLAG= 000050          T$TEST= 000033          T10     027506 G        T25     035076 G        WDPDR   023322 G
T$GMAN= 000000          T$TSTM= 177777          T11     027706 G        T26     035750 G        WORD1   002350 G
T$HILI= 177777          T$TSTS= 000001          T12     030120 G        T27     036226 G        WTWLNC  023516 G
T$LAST= 000001          T$$AU = 010025          T13     030366 G        T3      025756 G        WTWLNS  023546 G
T$LOLI= 000000          T$$AUT= 010022          T14     030730 G        T4      026206 G        WTWLPR  023572 G
T$LSYM= 010000          T$$CLE= 010023          T15     031314 G        T5      026370 G        X$ALWA= 000000
T$LTNO= 000033          T$$DU = 010024          T16     031524 G        T6      026546 G        X$FALS= 000040
T$NEST= 177777          T$$HAR= 010061          T17     031740 G        T7      026750 G        X$OFF$= 000400
T$NS0 = 000000          T$$HW = 010000          T18     032204 G        T8      027142 G        X$TRUE= 000020
T$NS1 = 000005          T$$INI= 010021          T19     032454 G        T9      027334 G        $PATCH  036744 G
T$PTNU= 000000          T$$MSG= 010016

. ABS.  037020    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 269
Work  file writes: 279
Size of work file: 35712 Words  ( 140 Pages)
Size of core pool: 19714 Words  ( 75 Pages)
Operating  system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:52:56.11
CVDHADO.BIN,CVDHADO.LST/-SP=SVC40/ML,CVDHADO.P11
```