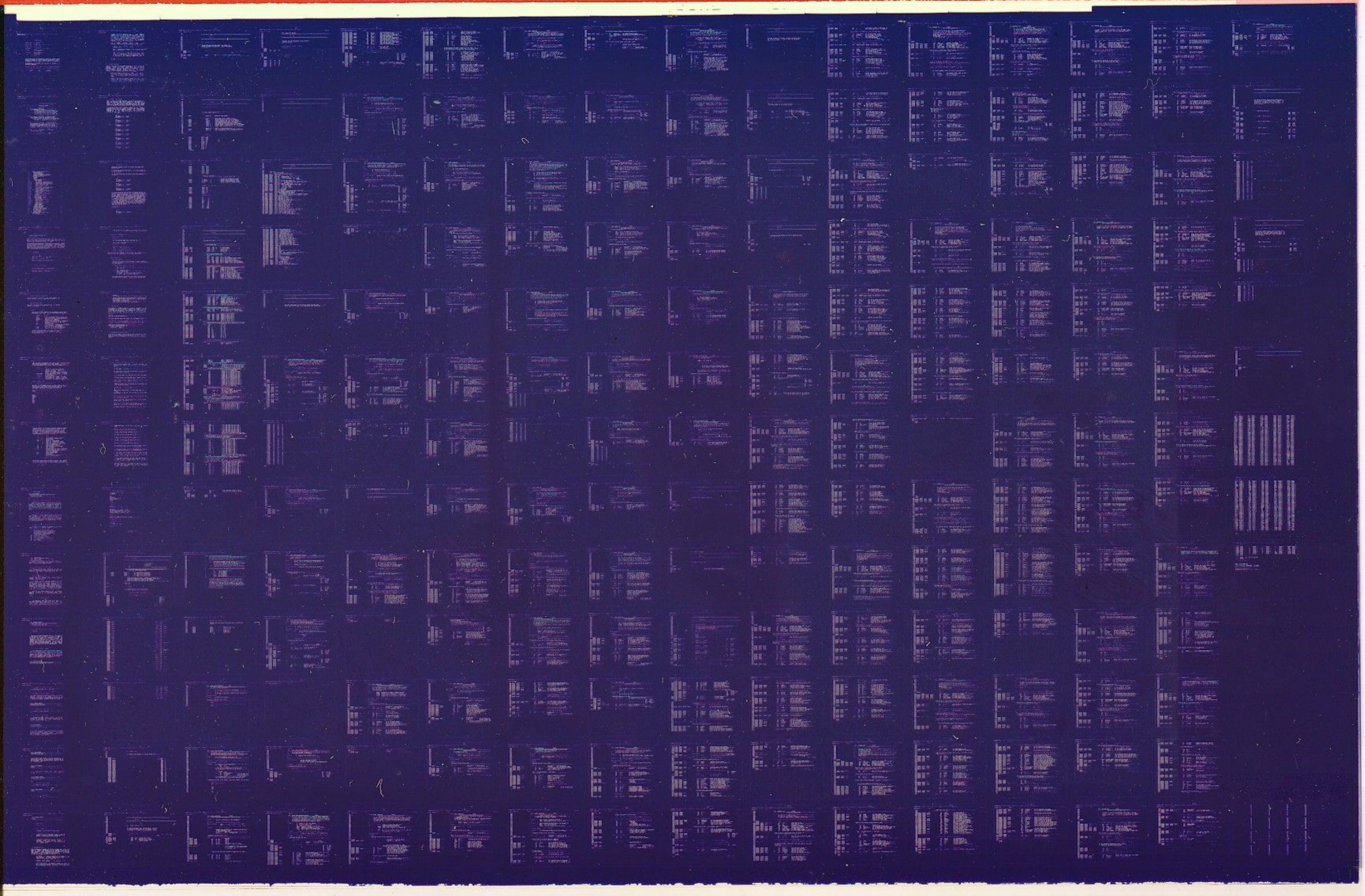
DHV11-M

DHV11-M FUNC TST PT 2
CVDHBD0

AH-T655D-MC
1 OF 1 JUL 1985
 MADE IN USA

digital



```
.REM &
```

IDENTIFICATION

PRODUCT CODE:	AC-T654D-MC
PRODUCT NAME:	CVDHBDO DHV11-M FUNC TST PART 2
PRODUCT DATE:	29 MARCH 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 tests from this program to CVDHA (Part 1):
                Old CVDHB (version A) tests 2 through 8 are
                now New CVDHA (version B) tests 20 through 26.
        Moved test from CVDHC (Part 3) into this program:
                Old CVDHC (version A) tests 4 through 6 are
                now New CVDHB (version B) tests 13 through 15.

Version C0              16-DEC-83       Bert Kleinschmidt
        Fixed typographical errors discovered to date.
        Added 2 tests to allow this program to be used for a
            complete checkout of cables and distribution panels.
                New test 24 - Tests continuity of TX/RX lines.
                New test 25 - Detects TX line interactions.

Version D0              17-Jul-84       Bert Kleinschmidt
        Fixed typographical errors discovered to date.
        Modified control of processor priority and LTC throughout
        the program to guarantees a less than 2 second
        responce to a Break request while running under APT.
        Fixed bug in IAUTO Active test which caused an XON
        character to arrive late on processors with cache.

Version D0              28-Sep-84       Peter ONeil
        Modified clean up section to turn off clock if clock
        had been turned on.

                        29-Mar-85       Howard Marshall
        Did formal release of version CVDHBDO.
```

TABLE OF CONTENTS

1.0 GENERAL PROGRAM CONSIDERATIONS

1.1 PROGRAM ABSTRACT

CVDHB is part two of the DHV11-M functional verification test. This part of the test verifies that the major communication functions of the board are functioning correctly. This program does not perform extensive data transmission and reception tests.

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:

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

1.3 RELATED DOCUMENTS AND STANDARDS

- DHV11-M Hardware Manual - This manual describes the functions and uses of the DHV11-M device.
- 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 is run.

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.

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.

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. With staggered loopback a pair of lines with the specified transmit line and another receive line will be tested. Therefore, to guarantee that both the transmitter and receiver of a specified line are tested when using the staggered loopback connector, both the intended line AND its mate must be selected (ie. to test line 1, select both line 1 and line 3). In nonstaggered testing, a bit in the active lines bit map selects the transmitter and receiver for the same line.

4. TYPE OF LOOPBACK (1=INTERNAL, 2=H3277, 3=H325) - This question requests the type of loopback to be used in testing the DHV11. The following types of loopback are supported:

   - o INTERNAL - Only internal UART loopback is to be used in testing the DHV.
   - o H3277 - Staggered Berg connector(s) are installed at the end of the 40 wire cables in place of the DHV11 distribution panels.
   - o H325 - Single line, 25 pin loopback connectors (type H325) are installed on the lines to be tested. These connectors can be installed on the distribution panel or on the end of the terminal or modem cable. The H325 connectors must have the removable jumpers installed.

5. 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 whether the program should report the number of the unit which it is testing as it begins to test each unit. The unit number will only be reported if more than one unit is being tested.

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.

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 (D) ?  8<CR>

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

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

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

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

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

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

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

UNIT 8
CSR ADDRESS (O)  160000<CR>
SUB-DEVICE # (O) ?  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 NOT 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:

```
CVDHB 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:

```
CVDHB 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 CVDHB:

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. DMA.START test - Verifies that each DMA.START bit will initiate a DMA TX on a line.

3. DMA.ABORT test - Verifies that the DMA.ABORT bit on each line will stop a DMA transmission and return a TX.ACTION and that the DMA can then be restarted.

4. O.AUTO inactive test - Verifies that the UUT will not respond to incoming XON and XOFF characters when O.AUTO is inactive.

5. O.AUTO active test - Verifies that the UUT respond correctly to incoming XON and XOFF characters when O.AUTO is active.

6. I.AUTO inactive test - Verifies that the UUT will not generate and TX XON or XOFF characters in response to the FIFO conditions if the I.AUTO bit is inactive.

7. I.AUTO active test - Verifies that the UUT will generate and TX XON and XOFF characters in response to the FIFO conditions if the I.AUTO bit is active.

8. FIFO data test - Verifies that the FIFO will hold 256 characters without corrupting data.

9. FIFO 3/4 level inactive test - Verifies that the FIFO 3/4 alarm does not become active until the FIFO becomes 3/4 full.

10. FIFO 3/4 level active test - Verifies that the FIFO 3/4 alarm becomes active, and remains active, when the FIFO is more than 3/4 full.

11. FIFO 3/4 level active/inactive test - Verifies that the FIFO 3/4 alarm, once activated, remains active until the FIFO level is reduced below 1/2.

12. FIFO 1/2 level test - Verifies that FIFO 1/2 level indicator becomes active and remains active as the FIFO level is reduced below the 1/2 full point.

13. BREAK test - Verifies that setting the BREAK bit on any line causes that line to go to a spacing state and that clearing the BREAK bit removes the line from the spacing state.

14. No OVERRUN.ERROR test - Verifies that the UUT will receive the maximum number of characters without causing an overrun error.

15. OVERRUN.ERROR test - Verifies that if more than the maximum number of characters are sent to the UUT overrun errors occur.

16. DTR test - Verifies that changing the UUT LNCTRL DTR bit affects the state of the DTR control line.

17. RTS test - Verifies that changing the UUT LNCTRL RTS bit affects the state of the RTS control line.

18. DSR test - Verifies that DSR status signal correctly reports the state of the looped back DTR control line.

19. RI test - Verifies that RI status signal correctly reports the state of the looped back DTR control line.

20. CTS test - Verifies that CTS status signal correctly reports the state of the looped back RTS control line.

21. DCD test - Verifies that DCD status signal correctly reports the state of the looped back RTS control line.

22. DTR interactions test - Verifies that changing the state of the DTR control signal on any line does not affect the state of any status signals that it is not looped back to.

23. RTS interactions test - Verifies that changing the state of the RTS control signal on any line does not affect the state of any status signals that it is not looped back to.

24. TX lines test - Verifies that each active line has continuity to the correct receive line. This test is only run if H325 or H3277 loopback is specified in the hardware questions dialogue. This test is intended to be used to verify cable and distribution panel data line connections.

25. TX lines interactions test - Verifies that each active TX line does not influence any RX lines (except the one that it is looped back to) or any modem status signals. This test is only run if H325 or H3277 loopback is specified in the hardware questions dialogue.

26. 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.

PROGRAM DOCUMENT

6.0 EXAMPLE ERROR FREE PASS

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

```
.R CVDHBDO
CVDHBDO.BIC

DRS
CVDHB-D-0
DHV11-M FUNC TST PART 2
UNIT IS DHV11-M
RESTART ADDR: 147670
DR>STA

CHANGE HW (L)  ? Y

# UNITS (D)  ? 2

UNIT 0
CSR ADDRESS:  (0)  160460 ? †Z

UNIT 1
CSR ADDRESS:  (0)  160460 ? 160040
INTERRUPT VECTOR ADDRESS:  (0) 300 ? 320
ACTIVE LINE BIT MAP:  (0)  377 ? <CR>
TYPE OF LOOPBACK (1=INTERNAL, 2=H3277, 3=H325):  (0)  2 ? 1
INTERRUPT BR LEVEL:  (0)  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

 TESTING UNIT :  0(D)

 TESTING UNIT :  1(D)

CVDHB EOP    1
    0 CUMULATIVE ERRORS

 TESTING UNIT :  0(D)
†C
DR> EXIT
```

.

&

```
1098                                    .LIST SEQ,LOC,BIN,MEB
1099                                    .NLIST CND
```

PROGRAM DOCUMENT

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


002000                          HEADER  CVDHB,D,0,22,0,PRI07
002000
002000  103                                                     L$NAME::
002001  126                                                             .ASCII /C/
002002  104                                                             .ASCII /V/
002003  110                                                             .ASCII /D/
002004  102                                                             .ASCII /H/
002005  000                                                             .ASCII /B/
002006  000                                                             .BYTE   0
002007  000                                                             .BYTE   0
002010                                                                  .BYTE   0
002010  104                                                     L$REV::
                                                                        .ASCII  /D/
002011                                                          L$DEPO::
```

Program Header SEQ 0023

```
002011     060
002012
002012  000000
002014
002014  000022
002016
002016  035724
002020
002020  036206
002022
002022  002212
002024
002024  002224
002026
002026  036446
002030
002030  000000
002032
002032  000000
002034
002034  000000
002036
002036  000000
002040
002040  002124
002042
002042  000340
002044
002044  000000
002046
002046  000000
002046
002046  000000
002050
002050     004
002051     000
002052
002052  000000
002054  000000
002056
002056  000000
002060
002060  004154
002062
002062  017064
002064
002064  000000
002066
002066  000000
002070
002070  000000
002072
002072  020012
002074
002074  000000
002076
002076  004164
002100
002100  104035
```

```
                .ASCII  /0/
L$UNIT::
                .WORD   0
L$TIML::
                .WORD   22
L$HPCP::
                .WORD   L$HARD
L$SPCP::
                .WORD   L$SOFT
L$HPTP::
                .WORD   L$HW
L$SPTP::
                .WORD   L$SW
L$LADP::
                .WORD   L$LAST
L$STA::
                .WORD   0
L$CO::
                .WORD   0
L$DTYP::
                .WORD   0
L$APT::
                .WORD   0
L$DTP::
                .WORD   L$DISPATCH
L$PRIO::
                .WORD   PRI07
L$ENVI::
                .WORD   0
L$EXP1::
                .WORD   0
L$MREV::
                .BYTE   C$REVISION
                .BYTE   C$EDIT
L$EF::
                .WORD   0
                .WORD   0
L$SPC::
                .WORD   0
L$DEVP::
                .WORD   L$DVTYP
L$REPP::
                .WORD   L$RPT
L$EXP4::
                .WORD   0
L$EXP5::
                .WORD   0
L$AUT::
                .WORD   0
L$DUT::
                .WORD   L$DU
L$LUN::
                .WORD   0
L$DESP::
                .WORD   L$DESC
L$LOAD::
                EMT     E$LOAD
```

L2

SEQ 0024

Program Header

```
                002102
                002102  004104
                002104
                002104  017100
                002106
                002106  017754
                002110
                002110  017752
                002112
                002112  017072
                002114
                002114  000000
                002116
                002116  000000
                002120
                002120  000000
   1162

                                                        L$ETP::
                                                                .WORD   L$ERRTBL
                                                        L$ICP::
                                                                .WORD   L$INIT
                                                        L$CCP::
                                                                .WORD   L$CLEAN
                                                        L$ACP::
                                                                .WORD   L$AUTO
                                                        L$PRT::
                                                                .WORD   L$PROT
                                                        L$TEST::
                                                                .WORD   0
                                                        L$DLY::
                                                                .WORD   0
                                                        L$HIME::
                                                                .WORD   0
```

DISPATCH TABLE

```
1174                                    .SBTTL  DISPATCH TABLE
1175
1176                                    ;++
1177                                    ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1178                                    ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1179                                    ;--
1180
1181 002122                                    DISPATCH 26
     002122  000032                                                    .WORD   26
     002124                                                  L$DISPATCH::
     002124  020126                                                    .WORD   T1
     002126  020416                                                    .WORD   T2
     002130  021004                                                    .WORD   T3
     002132  021424                                                    .WORD   T4
     002134  022164                                                    .WORD   T5
     002136  022724                                                    .WORD   T6
     002140  023320                                                    .WORD   T7
     002142  023742                                                    .WORD   T8
     002144  024232                                                    .WORD   T9
     002146  024534                                                    .WORD   T10
     002150  025226                                                    .WORD   T11
     002152  025716                                                    .WORD   T12
     002154  026362                                                    .WORD   T13
     002156  027006                                                    .WORD   T14
     002160  027506                                                    .WORD   T15
     002162  030434                                                    .WORD   T16
     002164  031134                                                    .WORD   T17
     002166  031634                                                    .WORD   T18
     002170  032250                                                    .WORD   T19
     002172  032664                                                    .WORD   T20
     002174  033300                                                    .WORD   T21
     002176  033714                                                    .WORD   T22
     002200  034310                                                    .WORD   T23
     002202  034704                                                    .WORD   T24
     002204  035222                                                    .WORD   T25
     002206  035642                                                    .WORD   T26
1182
```

```
   1190
   1191                         ;******************************************************************************
   1192                         ;
   1193                         ;               VDHB.DHT
   1194                         ;
   1195                         ;******************************************************************************
********
   1196
   1197
   1198
   1199                         .SBTTL  DEFAULT HARDWARE P-TABLE
   1200
   1201                         ;++
   1202                         ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
   1203                         ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
   1204                         ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
   1205                         ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
   1206                         ;--
   1207
   1208 002210                         BGNHW   DFPTBL
        002210  000004                                                          .WORD   L10000-L$HW/2
        002212                                                          L$HW::
        002212                                                          DFPTBL::
   1209
   1210 002212  160460                  .WORD   160460  ;Default CSR Address
   1211 002214  000300                  .WORD   300     ;Default Vector Address
   1212 002216  000377                  .WORD   MAPLNS  ;Default Active lines bit map
   1213 002220     002                  .BYTE   2       ;Default Loopback mode
   1214 002221     004                  .BYTE   4       ;Default BR Level
   1215
   1216 002222                          ENDHW
        002222                                                          L10000:
```

DEFAULT HARDWARE P-TABLE

```
                        ;*****************************************************************************
*
                        ;
                        ;               VDHB.SWT
                        ;
                        ;*****************************************************************************


                        .SBTTL  SOFTWARE P-TABLE

                        ;++
                        ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
                        ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
                        ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
                        ; AT RUN TIME.
                        ;--

002222                          BGNSW   SFPTBL
002222  000002                                                         .WORD   L10001-L$SW/2
002224                                                          L$SW::
002224                                                          SFPTBL::

002224  000020          OPTION::        .WORD   20      ;bit map of program control flags
002226  000000          NDERPT::        .WORD   0       ;Default number of individual data errors to rpt.

002230                          ENDSW
002230                                                          L10001:
```

```

                                ;*****************************************************************************
%
                                ;
                                ;           VDHB.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                          CSRO==0         ;CSR REGISTER OFFSET FROM THE CSR ADDRESS
        000002                          RBUFO==2        ;RECEIVE REGISTER OFFSET FROM THE CSR ADDRESS
        000002                          TXCHRO==2       ;TRANSMIT REGISTER OFFSET FROM THE CSR ADDRESS
        000004                          LPRO==4         ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
        000006                          STATO==6        ;STATUS REGISTER OFFSET FROM THE CSR ADDRESS
        000010                          LNCTRO==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                          TXBFCO==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.


 002230                                 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

```
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
```

```
                        ;******************************************************************************
                        ;
                        ;               VDHB.GDT
                        ;
                        ;******************************************************************************


                        .SBTTL  GLOBAL DATA SECTION

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

                        ;*****************************************************************************
                        ;       Unit Varible Area
                        ;*****************************************************************************

002230  000300                  RXVECA:: .WORD  300      ;RX VECTOR ADDRESS.
002232  000304                  TXVECA:: .WORD  304      ;TX VECTOR ADDRESS.
002234  000377                  ACTLNS:: .WORD  377      ;ACTIVE LINE BIT MAP.
002236     000                  LOPBCK:: .BYTE  0        ;LOOPBACK MODE
002237     004                  BRLEVL:: .BYTE  4        ;INTERRUPT BUS REQUEST LEVEL
002240  000000                  UNITN:: .WORD   0        ;UNIT NUMBER.

                        ;*****************************************************************************
                        ;       Device Register Address Table
                        ;*****************************************************************************
002242                          DRADRT::
002242  160000                  CSRA::   .WORD   160000  ;DHV11-M CSR ADDRESS
002244  160002          TXCHA:: RBUFA:: .WORD   160002  ;DHV11-M RECEIVE/TRANSMIT BUFFER ADDRESS
002246  160004                  LPRA::   .WORD   160004  ;DHV11-M LINE PARAMETER REGISTER ADDRESS
002250  160006                  STATA:: .WORD   160006  ;DHV11-M STATUS REGISTER ADDRESS
002252  160010                  LNCTRA:: .WORD  160010  ;DHV11-M LINE CONTROL REGISTER ADDRESS
002254  160012                  TXAD1A:: .WORD  160012  ;DHV11-M TRANSMIT BUFFER 1 REGISTER ADDRESS
002256  160014                  TXAD2A:: .WORD  160014  ;DHV11-M TRANSMIT BUFFER 2 REGISTER ADDRESS
002260  160016                  TXBFCA:: .WORD  160016  ;DHV11-M TRANSMIT BUFFER COUNT REGISTER ADDRESS

                        ;*****************************************************************************
                        ;       Assorted global variables:
                        ;*****************************************************************************
002262  000000                  BUFPTR:: .WORD  0        ;STORAGE FOR RECEIVE CHARACTER BUFFER POINTER.
002264  000000                  CTRLCF:: .WORD  0        ;STORAGE FOR THE CONTROL-C FLAG.
002266  000001                  TSTNUM:: .WORD  1        ;STORAGE FOR THE TEST NUMBER.
002270  000000                  IESTAT:: .WORD  0        ;STORAGE FOR STATES OF THE DUT INT ENABLE BITS.
002272  031463                  LGRP1M:: .WORD  31463    ;BIT MAP OF LINES IN LINE GROUP I.
002274  146314                  LGRP2M:: .WORD  146314   ;BIT MAP OF LINES IN LINE GROUP II.
002276  000000                  PASCNT:: .WORD  0        ;STO'G FOR PASS COUNT USED IN ROM VERSION# TST.
002300  000000                  RXINTC:: .WORD  0        ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
002302  000000                  RXINTF:: .WORD  0        ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
002304  000000                  TXINTC:: .WORD  0        ;STORAGE FOR TRANSMIT INTERRUPT COUNT.
002306  000000                  TXINTF:: .WORD  0        ;STORAGE FOR TRANSMIT INTERRUPT FLAGS.
002310  000000                  TP4VEC:: .WORD  0        ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
002312  000000                  TP4FLG:: .WORD  0        ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
002314  000000                  WORD1::  .WORD  0        ;LOCATION FOR PASSING INDIRECT PARAMETERS.
```

```
                        ;*****************************************************************************
                        ;       Line Time Clock variables and storage.
                        ;*****************************************************************************
002316  177546                CLKCSR:: .WORD  177546  ;CSR ADDRESS OF THE LTC.
002320  000300                CLKBRL:: .WORD  PRI06   ;INTERRUPT PRIORITY LEVEL OF THE LTC.
002322  000100                CLKVEC:: .WORD  100     ;INTERRUPT VECTOR ADDRESS OF THE LTC.
002324  000074                CLKHRZ:: .WORD  60.     ;INTERRUPT FREQUENCY OF THE LTC.
002326  000000                TIMER1:: .WORD  0       ;HARDWARE CLOCK COUNTER #1.
002330  000000                TIMER2:: .WORD  0       ;HARDWARE CLOCK COUNTER #2.
002332  000170                TIMER3:: .WORD  120.    ;HARDWARE BREAK COUNTER LOCATION.
002334  000170                BCOUNT:: .WORD  120.    ;BREAK COUNT VALUE IN CLOCK TICKS.
002336  000021                MSTICK:: .WORD  17.     ;NUMBER OF MILLI-SECONDS PER LTC TICK.
002340  000062                MSLCNT:: .WORD  62      ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.

                        ;*****************************************************************************
                        ;       Memmory Management Variables and Flags.
                        ;*****************************************************************************
002342  177572                MMSR0:: .WORD   177572  ;ADDRESS OF MEM MGT STATUS REGISTER #0.
002344  000000                MMPRES:: .WORD  0       ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
002346  000000                MMENAB:: .WORD  0       ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).

002350                  PARATB::                        ;BASE OF MEM MGT PAR ADDRESS TABLE.
002350  172340                PAR0A:: .WORD   172340  ;ADDRESS OF MEM MGT PAR #0.
002352  172342                PAR1A:: .WORD   172342  ;ADDRESS OF MEM MGT PAR #1.
002354  172344                PAR2A:: .WORD   172344  ;ADDRESS OF MEM MGT PAR #2.
002356  172346                PAR3A:: .WORD   172346  ;ADDRESS OF MEM MGT PAR #3.
002360  172350                PAR4A:: .WORD   172350  ;ADDRESS OF MEM MGT PAR #4.
002362  172352                PAR5A:: .WORD   172352  ;ADDRESS OF MEM MGT PAR #5.
002364  172354                PAR6A:: .WORD   172354  ;ADDRESS OF MEM MGT PAR #6.
002366  172356                PAR7A:: .WORD   172356  ;ADDRESS OF MEM MGT PAR #7.
002370                  PARATE::                        ;END OF PAR ADDRESS TABLE.
                        ;*****************************************************************************
                        ;       Table of words with corresponding bit set for generation of bit maps.
                        ;*****************************************************************************
002370  000001          BITTBL:: .WORD  1               ;BIT 0 SET.
002372  000002                   .WORD  2               ;BIT 1 SET.
002374  000004                   .WORD  4               ;BIT 2 SET.
002376  000010                   .WORD  10              ;BIT 3 SET.
002400  000020                   .WORD  20              ;BIT 4 SET.
002402  000040                   .WORD  40              ;BIT 5 SET.
002404  000100                   .WORD  100             ;BIT 6 SET.
002406  000200                   .WORD  200             ;BIT 7 SET.
002410  000400                   .WORD  400             ;BIT 8 SET.
002412  001000                   .WORD  1000            ;BIT 9 SET.
002414  002000                   .WORD  2000            ;BIT 10 SET.
002416  004000                   .WORD  4000            ;BIT 11 SET.
002420  010000                   .WORD  10000           ;BIT 12 SET.
002422  020000                   .WORD  20000           ;BIT 13 SET.
002424  040000                   .WORD  40000           ;BIT 14 SET.
002426  100000                   .WORD  100000          ;BIT 15 SET.

                        ;*****************************************************************************
                        ;*      GPR Save Areas Zero and One.
                        ;*****************************************************************************
002430                  GPRS0B::                        ;BASE OF GPR SAVE AREA NUMBER ZERO.
002430  000000                   .WORD  0               ;WORD 1, STORAGE FOR R1.
002432  000000                   .WORD  0               ;WORD 2, STORAGE FOR R2.
```

GLOBAL DATA SECTION

```
1421 002434  000000                .WORD   0               ;WORD 3, STORAGE FOR R3.
1422 002436  000000                .WORD   0               ;WORD 4, STORAGE FOR R4.
1423 002440  000000                .WORD   0               ;WORD 5, STORAGE FOR R5.
1424                        ;*******************************************************************************
1425                        ;       Storage area for the BMP code queue.
1426                        ;*******************************************************************************
1427 002442  000000        BMPCQP::        .WORD   0       ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
1428 002444                BMPCQB::        .BLKW   64.     ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
1429 002644                BMPCQE::                        ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1430                        ;*******************************************************************************
1431                        ;       Storage area for the contents of the DUT STAT register states.
1432                        ;*******************************************************************************
1433 002644                STSTB::                         ;BASE OF DUT STAT STORAGE TABLE.
1434 002644  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 0.
1435 002646  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 1.
1436 002650  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 2.
1437 002652  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 3.
1438 002654  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 4.
1439 002656  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 5.
1440 002660  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 6.
1441 002662  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 7.
1442 002664  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 8.
1443 002666  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 9.
1444 002670  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 10.
1445 002672  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 11.
1446 002674  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 12.
1447 002676  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 13.
1448 002700  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 14.
1449 002702  000000                .WORD   0               ;STORAGE FOR STAT REGISTER FOR LINE 15.
1450 002704                STSTE::                         ;END OF DUT STAT STORAGE TABLE.
1451                        ;*******************************************************************************
1452                        ;       General table and buffer area--513 words.
1453                        ;*******************************************************************************
1454 002704                BUFBAS::                        ;BASE OF MEMORY BUFFER.
1455 002704                ERLTBL::        .BLKW   128.    ;FIRST HALF OF GENERAL TABLE OR BUFFER.
1456 003304                BUFMID::        .BLKW   64.     ;SECOND HALF OF GENERAL TABLE OR BUFFER.
1457 003504                BUF3QT::        .BLKW   64.     ;LAST QUARTER OF THE BUFFER AREA.
1458 003704                BUFEND::                        ;END OF GENERAL PURPOSE MEMORY BUFFER.
1459 003704                ENDETB::        .BLKW   16.     ;BUFFER OVERFLOW SPACE.
1460                        ;*******************************************************************************
1461                        ;               Reception table of counters
1462                        ;*******************************************************************************
1463 003744                RXCNTB:: .BLKW  16.             ;RECEPTION CHARACTER COUNTERS TABLE.
1464                        ;*******************************************************************************
1465                        ;* Table for storage of RX/TX line number associations.
1466                        ;* The associations are stored as line number times 2 for use as offsets
1467                        ;*    when accessing a table of words.
1468                        ;*    NOTE: Do not write a non-zero value into the upper byte of any entry.
1469                        ;*******************************************************************************
1470 004004                TXRXLB::                        ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1471 004004  000000                .WORD   0               ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
1472 004006  000002                .WORD   2.              ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
1473 004010  000004                .WORD   4.              ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
1474 004012  000006                .WORD   6.              ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
1475 004014  000010                .WORD   8.              ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
1476 004016  000012                .WORD   10.             ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
1477 004020  000014                .WORD   12.             ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
```

```
1478 004022  000016                    .WORD   14.             ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
1479 004024  000020                    .WORD   16.             ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
1480 004026  000022                    .WORD   18.             ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
1481 004030  000024                    .WORD   20.             ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
1482 004032  000026                    .WORD   22.             ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
1483 004034  000030                    .WORD   24.             ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
1484 004036  000032                    .WORD   26.             ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
1485 004040  000034                    .WORD   28.             ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
1486 004042  000036                    .WORD   30.             ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
1487 004044                    TXRXLE::                        ;END OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1488                                   .EVEN                   ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1489                           ;*************************************************************************
1490                           ;* Table for storage of RX/TX line number associations.
1491                           ;* The associations are stored as line numbers which can be used as such or
1492                           ;*    as offsets when accessing a table of bytes.
1493                           ;*************************************************************************
1494 004044                    TXRLNB::                        ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1495 004044       000                 .BYTE   0               ;TX/RX LINE FOR RX/TX LINE 0.
1496 004045       001                 .BYTE   1.              ;TX/RX LINE FOR RX/TX LINE 1.
1497 004046       002                 .BYTE   2               ;TX/RX LINE FOR RX/TX LINE 2.
1498 004047       003                 .BYTE   3.              ;TX/RX LINE FOR RX/TX LINE 3.
1499 004050       004                 .BYTE   4.              ;TX/RX LINE FOR RX/TX LINE 4.
1500 004051       005                 .BYTE   5.              ;TX/RX LINE FOR RX/TX LINE 5.
1501 004052       006                 .BYTE   6.              ;TX/RX LINE FOR RX/TX LINE 6.
1502 004053       007                 .BYTE   7.              ;TX/RX LINE FOR RX/TX LINE 7.
1503 004054       010                 .BYTE   8.              ;TX/RX LINE FOR RX/TX LINE 8.
1504 004055       011                 .BYTE   9.              ;TX/RX LINE FOR RX/TX LINE 9.
1505 004056       012                 .BYTE   10.             ;TX/RX LINE FOR RX/TX LINE 10.
1506 004057       013                 .BYTE   11.             ;TX/RX LINE FOR RX/TX LINE 11.
1507 004060       014                 .BYTE   12.             ;TX/RX LINE FOR RX/TX LINE 12.
1508 004061       015                 .BYTE   13.             ;TX/RX LINE FOR RX/TX LINE 13.
1509 004062       016                 .BYTE   14.             ;TX/RX LINE FOR RX/TX LINE 14.
1510 004063       017                 .BYTE   15.             ;TX/RX LINE FOR RX/TX LINE 15.
1511 004064                    TXRLNE::                        ;END OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1512                                   .EVEN                   ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1513                           ;*************************************************************************
1514                           ;* Table of TX/RX line number associations in staggered loopback.
1515                           ;* The associations are stored as line number times 2 for use as offsets
1516                           ;*    when accessing a table of words.
1517                           ;* This is a table of data for reading only.  Use to load the above table.
1518                           ;*    NOTE: Must convert from BYTEs to WORDs when loading above table.
1519                           ;*************************************************************************
1520 004064                    STGTRB::                        ;BASE OF STAGGERED TX/RX LINE NUMBER TABLE.
1521 004064       004                 .BYTE   4.              ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
1522 004065       006                 .BYTE   6.              ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
1523 004066       000                 .BYTE   0               ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
1524 004067       002                 .BYTE   2.              ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
1525 004070       014                 .BYTE   12.             ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
1526 004071       016                 .BYTE   14.             ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
1527 004072       010                 .BYTE   8.              ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
1528 004073       012                 .BYTE   10.             ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
1529 004074       024                 .BYTE   20.             ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
1530 004075       026                 .BYTE   22.             ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
1531 004076       020                 .BYTE   16.             ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
1532 004077       022                 .BYTE   18.             ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
1533 004100       034                 .BYTE   28.             ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
1534 004101       036                 .BYTE   30.             ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
```

GLOBAL DATA SECTION

```
1535 004102     030                         .BYTE   24.              ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
1536 004103     032                         .BYTE   26.              ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
1537                                        .EVEN                    ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1550 004104                                ERRTBL
     004104                                                                 L$ERRTBL::
     004104  000000              ERRTYP::          .WORD   0
     004106  000000              ERRNBR::          .WORD   0
     004110  000000              ERRMSG::          .WORD   0
     004112  000000              ERRBLK::          .WORD   0
1551
1552                                       .EVEN
```

GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.

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

GPR FRAME ACCESS EQUATES

```
                                .SBTTL GPR FRAME ACCESS EQUATES
                                ;+++
                                ;Equates that allow access to the stack frame.  These are the
                                ;offsets into the stack for registers saved during the PREG05
                                ;routine.
                                ;---

000036                          LPCSLT==        36      ;Offset for last return PC.
000016                          PCSLOT==        16      ;Offset for return PC.
000014                          R5SLOT==        14      ;Offset for R5.
000012                          R4SLOT==        12      ;Offset for R4.
000010                          R3SLOT==        10      ;Offset for R3.
000006                          R2SLOT==        6       ;Offset for R2.
000004                          R1SLOT==        4       ;Offset for R1.
000002                          R0SLOT==        2       ;Offset for R0.
```

GLOBAL MACRO DEFINITION - SAVE -

```
                        .SBTTL  GLOBAL MACRO DEFINITION         - SAVE -
                        ;*************************************************************************
                        ;*      This macro is used at the beginning of a subroutine to save the
                        ;*      contents of the GPRs R0 thru R5.
                        ;*
                        ;* INPUTS:      SP - Unchanged since subroutine was entered
                        ;*              R5SLOT - Offset to stack slot for R5 (Equated to 14 Octal)
                        ;*
                        ;* OUTPUTS:     GPR save area on the stack is loaded with the contents of GPRs
                        ;*              TOP OF STACK - Loaded with the return address into PREG05
                        ;*
                        ;* CALLING SEQUENCE:    SAVE
                        ;*
                        ;* COMMENTS:    No arguments are allowed.
                        ;*              The PASS macro should be called to restore the GPR values.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: PREG05.
                        ;*************************************************************************

                                .MACRO  SAVE
                                .LIST
                                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                .NLIST
                                .ENDM   SAVE
```

GLOBAL MACRO DEFINITION - PASS -

```
                        .SBTTL  GLOBAL MACRO DEFINITION          - PASS -
                        ;*************************************************************************
                        ;*      This macro is used in conjunction with the SAVE macro.  It is
                        ;*      called at end of a subroutine to pass parameters in GPRs back to the
                        ;*      calling routine by altering the GPR save area on the stack and then
                        ;*      returning to PREG05 to restore the GPRs to their saved values.
                        ;*
                        ;* INPUTS:      Only allowed ARGUMENTS are "R0" thru "R5".
                        ;*              R0SLOT thru R5SLOT must be equated to their respective GPR save
                        ;*                      slot offsets before calling this macro.
                        ;*
                        ;* OUTPUTS:     The GPR values are put in their respective slots on the stack.
                        ;*
                        ;* CALLING SEQUENCE:    PASS    R0,R1,...
                        ;*
                        ;* COMMENTS:    Any combination of GPR arguments may be listed in any order.
                        ;*              For example, the following are legal:
                        ;*                      PASS    R1
                        ;*                      PASS    R4,R0,R2
                        ;*              The GPRs listed as arguments will be passed intact to the
                        ;*              calling routine, all other GPRs will be restored.
                        ;*              The SP must be at its original value when PASS is called.
                        ;*
                        ;*              The macro call
                        ;*                      PASS    R0,R3
                        ;*              expands into the following assembly code:
                        ;*                      MOV     R0,R0SLOT(SP)           ;PUT R0 IN STACK SLOT.
                        ;*                      MOV     R3,R3SLOT(SP)           ;PUT R3 IN STACK SLOT.
                        ;*                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                        ;*              In this example GPRs R1, R2, R4, and R5 will be restored to
                        ;*              their values contained in the stack frame and R0 and R3
                        ;*              will be left at their values prior to this PASS call.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: (PREGRT - Label within PREG05, value on stack.)
                        ;*************************************************************************

                                .MACRO  PASS    A,B,C,D,E,F
                                .IRP    X,<A,B,C,D,E,F>
                                .IF     NB,X
                                .LIST
                                        MOV     X,X'SLOT(SP)            ;PUT X IN STACK SLOT.
                                .NLIST
                                .ENDC
                                .ENDM
                                .LIST
                                        JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                .NLIST
                                .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,ROSLOT(SP)        ]
                                ;*              Make sure the SP is at its original value when you do this.
                                ;*
                                ;*SUBORDINATE ROUTINES CALLED:  None.
                                ;*******************************************************************************
004114                          PREG05:                 ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
004114  010446                          MOV     R4,-(SP)        ;SAVE R4
004116  010346                          MOV     R3,-(SP)        ;SAVE R3
004120  010246                          MOV     R2,-(SP)        ;SAVE R2
004122  010146                          MOV     R1,-(SP)        ;SAVE R1
004124  010046                          MOV     R0,-(SP)        ;SAVE R0
004126  010546                          MOV     R5,-(SP)        ;PUSH RETURN PC ON TOP OF STACK
004130  016605  000014                  MOV     R5SLOT(SP),R5   ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS
                
004134  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.
                                        ;---
                
004136  012605                  PREGRT:: MOV    (SP)+,R5        ;Put return PC in R5.
004140  012600                          MOV     (SP)+,R0        ;Restore R0.
004142  012601                          MOV     (SP)+,R1        ;Restore R1.
004144  012602                          MOV     (SP)+,R2        ;Restore R2.
004146  012603                          MOV     (SP)+,R3        ;Restore R3.
004150  012604                          MOV     (SP)+,R4        ;Restore R4.
                
004152  000205                          RTS     R5              ;Return to the subroutine which called PREG05.
                                                                  ;restoring R5 in the process.
```

GLOBAL TEXT SECTION

```
                                .SBTTL  GLOBAL TEXT SECTION
                                ;*********************************************************************************************
                                ;
                                ;               SKL1.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
                                ;
004154                                  DEVTYP  <DHV11-M>
004154                                                                  L$DVTYP::
004154  104     110     126                                                     .ASCIZ  %DHV11-M%
004157  061     061     055
004162  115     000
                                                                                .EVEN


                                ; TEST DESCRIPTION
                                ;
004164                                  DESCRIPT        <DHV11-M FUNC TEST PART 2>
004164                                                                  L$DESC::
004164  104     110     126                                                     .ASCIZ  /DHV11-M FUNC TEST P
ART 2/
004167  061     061     055
004172  115     040     106
004175  125     116     103
004200  040     124     105
004203  123     124     040
004206  120     101     122
004211  124     040     062
004214  000
                                                                                .EVEN
                                .EVEN
```

GLOBAL TEXT SECTION

```
   1776
   1777                         ;*****************************************************************************************************
*
   1778                         ;
   1779                         ;                  VDHB.FMT
   1780                         ;
   1781                         ;*****************************************************************************************************
   1782
   1783
   1784
   1785                         ;
   1786                         ; FORMAT STATEMENTS USED IN PRINT CALLS
   1787                         ;
   1788
   1799
   1800
```

```
                                         ;*********************************************************************
                                         ;
                                         ;                VDHB.MSG
                                         ;
                                         ;**********************************************************************
**


               .NLIST BIN
               .SBTTL  GLOBAL MESSAGE AREA
                       ;*********** FORMAT STATEMENTS ***********
       004216  MFUNIT:: .ASCIZ /*N*A TESTING UNIT :*D4*N/
       004247  EF0503:: .ASCIZ /*T*N/
       004254  EF0505:: .ASCIZ /*A     *D5*A ILLEGAL INTERRUPTS RECEIVED.*N/
       004327  EF1601:: .ASCIZ /*A   *T*A ABORTED *N/
       004353  EF3001:: .ASCIZ /*A    EXPECTED OR CORRECT VALUE: *O3*N/
       004422  EF3002:: .ASCIZ /*A    ACTUAL OR MEASURED  VALUE: *O3*N/
       004471  EF6401:: .ASCIZ /*A                                 *D2*A(D).*N/
       004546  EF7801:: .ASCIZ /*T*A ON LINE *D2*A DECIMAL.*N/
       004604  EF8401:: .ASCIZ /*A  *T*A FOR LINE *D2*A(D) AFFECTS OTHER MODEM SIGNALS.*N/
       004676  EF8402:: .ASCII /*A        CHANGING *T*A FOR LINE *D2*A(D) AFFECTED /
       004761           .ASCIZ /*T*A FOR LINE *D2*A(D).*N/
       005013  EF9001:: .ASCIZ /*A  UNEXPECTED *T*A FOUND IN RECEIVE CHAR FIFO:*N/
       005075  EF9002:: .ASCIZ /*A     CODE IS ASSOCIATED WITH LINE: *D2*N/
       005147  EF9003:: .ASCIZ /*A        CODE IS: *O3*N/
       005176  EF9004:: .ASCIZ /*A     *T*A VALUE: *O3*N/
       005226  EF9005:: .ASCIZ /*A     *T*A VALUE: NONE*N/
       005257  EF9006:: .ASCIZ /*A  *T*A *D2*N/
       005276  EF9019:: .ASCIZ /*A  *T*A *O6*N/
       005315  EF9301:: .ASCIZ /*A *T*D2*A(D),  BMP CODE REPORTED :*O3*A(O)*N/
       005373  EF9302:: .ASCIZ /*A  OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)*N/
                       ;************ MESSAGE AREA ************
       005473  EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
       005531  EM0525:: .ASCIZ /  RX INTERRUPT(S) RECEIVED WITH RX INTERRUPTS DISABLED./
       005621  EM0526:: .ASCIZ /  TX INTERRUPT(S) RECEIVED WITH TX INTERRUPTS DISABLED./
       005711  EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
       005774  EM4001:: .ASCIZ /DMA_START BIT TEST/
       006017  EM4002:: .ASCIZ /DMA_START BIT BAD ON LINE: /
       006053  EM4101:: .ASCIZ /DMA_ABORT BIT TEST/
       006076  EM4102:: .ASCIZ /DMA_ABORT BIT BAD ON LINE: /
       006132  EM4103:: .ASCIZ /DMA_START BIT FOUND SET AFTER DMA ABORTED ON LINE: /
       006216  EM4901:: .ASCIZ /OAUTO (INACTIVE) BIT TEST/
       006250  EM4902:: .ASCIZ /  OAUTO BIT BAD ON LINE: /
       006302  EM5001:: .ASCIZ /OAUTO (ACTIVE) BIT TEST/
       006332  EM5101:: .ASCIZ /IAUTO (INACTIVE) TEST/
       006360  EM5102:: .ASCIZ /IAUTO BIT FOUND SET ON LINE: /
       006416  EM5103:: .ASCIZ /IAUTO BIT BAD ON LINE: /
       006446  EM5201:: .ASCIZ /IAUTO (ACTIVE) TEST/
       006472  EM5202:: .ASCIZ /IAUTO BIT FOUND CLR ON LINE: /
       006530  EM5301:: .ASCIZ /FIFO VALID DATA TEST/
       006555  EM5302:: .ASCIZ /FIFO BAD DATA FIELD CORRUPTED, TEST USED LINE:/
       006634  EM5303:: .ASCIZ /BMP CODE FOUND IN FIFO, TEST INVAILDATED/
       006705  EM5401:: .ASCIZ \FIFO 3/4 ALARM (INACTIVE) TEST\
       006744  EM5402:: .ASCIZ /FIFO BAD, ALARM SIGNAL DEFECTIVE/
       007005  EM5501:: .ASCIZ \FIFO 3/4 ALARM (ACTIVE) TEST\
       007042  EM5601:: .ASCIZ \FIFO 3/4 ALARM (ACTIVE/INACTIVE) TEST\
       007110  EM5701:: .ASCIZ \FIFO 1/2 LEVEL (ACTIVE/INACTIVE) TEST\
```

```
1866 007156  EM6401:: .ASCIZ /BREAK GENERATION TEST /
1867 007205  EM6402:: .ASCIZ /  BREAK NOT RECEIVED ON LINE(S):/
1868 007246  EM6601:: .ASCIZ /NO OVERRUN ERROR TEST/
1869 007274  EM6602:: .ASCIZ /  OVERRUN ERROR REPORTED WHEN NONE FORCED/
1870 007346  EM6701:: .ASCIZ /OVERRUN ERROR TEST/
1871 007371  EM6702:: .ASCIZ /  NO OVERRURN ERROR REPORTED, OVERRUN FORCED/
1872 007446  EM7801:: .ASCIZ /MODEM CONTROL DTR BIT TEST/
1873 007501  EM7802:: .ASCIZ / DTR BIT FAULTY ON LINE:/
1874 007532  EM7901:: .ASCIZ /MODEM CONTROL RTS BIT TEST/
1875 007565  EM7902:: .ASCIZ / RTS BIT FAULTY ON LINE:/
1876 007616  EM8001:: .ASCIZ /DSR MODEM STATUS SIGNAL TEST /
1877 007654  EM8002:: .ASCIZ /  DSR MODEM STATUS SIGNAL DEFECTIVE/
1878 007720  EM8101:: .ASCIZ /RI MODEM STATUS SIGNAL TEST /
1879 007755  EM8102:: .ASCIZ /  RI MODEM STATUS SIGNAL DEFECTIVE/
1880 010020  EM8201:: .ASCIZ /CTS MODEM STATUS SIGNAL TEST /
1881 010056  EM8202:: .ASCIZ /  CTS MODEM STATUS SIGNAL DEFECTIVE/
1882 010122  EM8301:: .ASCIZ /DCD MODEM STATUS SIGNAL TEST /
1883 010160  EM8302:: .ASCIZ /  DCD MODEM STATUS SIGNAL DEFECTIVE/
1884 010224  EM8401:: .ASCIZ /DTR MODEM CONTROL SIGNAL INTERACTIONS TEST /
1885 010300  EM8402:: .ASCIZ /DTR/
1886 010304  EM8403:: .ASCIZ /DSR/
1887 010310  EM8404:: .ASCIZ /RI/
1888 010313  EM8405:: .ASCIZ /DCD/
1889 010317  EM8406:: .ASCIZ /CTS/
1890 010323  EM8501:: .ASCIZ /RTS MODEM CONTROL SIGNAL INTERACTIONS TEST /
1891 010377  EM8502:: .ASCIZ /RTS/
1892 010403  EM8601:: .ASCIZ /CABLE, PANEL, LOOPBACK CONNECTOR DATA LINE TEST /
1893 010464  EM8602:: .ASCIZ \TX/RX LINE ERROR ON THE FOLLOWING LOOPED BACK TX LINES:\
1894 010554  EM8701:: .ASCIZ /TX DATA LINE INTERACTIONS TEST /
1895 010614  EM8702:: .ASCIZ /DATA LINE INTERACTIONS/
1896 010643  EM8703:: .ASCIZ /TX/
1897 010646  EM9009:: .ASCIZ /EXPECTED OR CORRECT/
1898 010672  EM9010:: .ASCIZ /ACTUAL OR MEASURED /
1899 010716  EM9017:: .ASCII /  FIFO WILL NOT PURGE (DATA.VALID STUCK SET),/
1900 010773          .ASCIZ / REMAINDER OF TEST SKIPPED./
1901 011027  EM9026:: .ASCIZ /     LPR CONTENTS: /
1902 011053  EM9104:: .ASCIZ /  UNEXPECTED DATA FOUND IN FIFO FROM LINE: /
1903 011127  EM9301:: .ASCIZ /BMP CODE REPORT/
1904 011147  EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
1905 011177  EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
1906 011244  EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
1907
1908                 .EVEN
1909                         .LIST BIN
```

GLOBAL MESSAGE AREA

```
1911
1912                            ;*****************************************************************************
1913                            ;
1914                            ;               SKL2.P11
1915                            ;
1916                            ;*****************************************************************************
1917
1918
1919
1920                            .SBTTL  GLOBAL ERROR REPORT SECTION
1921
1922                            ;++
1923                            ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
1924                            ; USED BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
1925                            ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
1926                            ;--
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0101 -
                        ;*******************************************************************************
                        ;*      This is an error reporting subroutine which prints additional error
                        ;*      information if an error is detected in TEST 1 (Register Address
                        ;*      Access Test).  This subroutine reports the type of access (Read or
                        ;*      Write or both) which caused a bus time-out trap (004 trap).
                        ;*      A message indicating that the DHV may be at the wrong Q-bus address
                        ;*      is also printed.
                        ;*
                        ;* INPUTS:      R5 - Error flag word.
                        ;*                      If bit 0 is set, a read error occured.
                        ;*                      If bit 1 is set, a write error occured.
                        ;*
                        ;* OUTPUTS:     Messages are printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER0101" as the message pointer
                        ;*                      parameter in the DRS error report macro call.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*******************************************************************************

011320                          BGNMSG  ER0101
011320                                                                  ER0101::
011320                          SAVE                    ;SAVE THE GPR CONTENTS.
011320  004567  172570                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

011324  032705  000001          BIT     #BIT0,R5        ;TEST FOR READ ERROR.
011330  001410                  BEQ     2$              ;SKIP READ ERROR MSG IF NO READ ERROR.
011332                          PRINTB  #MSG1           ;PRINT READ ERROR MESSAGE.
011332  012746  011424                                                  MOV     #MSG1,-(SP)
011336  012746  000001                                                  MOV     #1,-(SP)
011342  010600                                                          MOV     SP,R0
011344  104414                                                          TRAP    C$PNTB
011346  062706  000004                                                  ADD     #4,SP
011352  032705  000002          2$:     BIT     #BIT1,R5        ;TEST FOR WRITE ERROR.
011356  001410                  BEQ     4$              ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
011360                          PRINTB  #MSG2           ;PRINT WRITE ERROR MESSAGE.
011360  012746  011502                                                  MOV     #MSG2,-(SP)
011364  012746  000001                                                  MOV     #1,-(SP)
011370  010600                                                          MOV     SP,R0
011372  104414                                                          TRAP    C$PNTB
011374  062706  000004                                                  ADD     #4,SP
011400                          4$:     PRINTX  #MSG3           ;SUGGEST THAT DHV MAY BE AT WRONG ADDRESS.
011400  012746  011561                                                  MOV     #MSG3,-(SP)
011404  012746  000001                                                  MOV     #1,-(SP)
011410  010600                                                          MOV     SP,R0
011412  104415                                                          TRAP    C$PNTX
011414  062706  000004                                                  ADD     #4,SP
011420                          PASS                    ;RESTORE THE GPR CONTENTS.
011420  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
011422                          ENDMSG
011422                                                                  L10002:
011422  104423                                                          TRAP    C$MSG

011424     045     101     102  MSG1::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.%N/
```

GLOBAL ERROR REPORTING ROUTINE - ER0101 -

```
        011427  125     123     040
        011432  124     111     115
        011435  105     055     117
        011440  125     124     040
        011443  124     122     101
        011446  120     040     103
        011451  101     125     123
        011454  105     104     040
        011457  102     131     040
        011462  122     105     101
        011465  104     040     101
        011470  124     124     105
        011473  115     120     124
        011476  056     045     116
        011501  000
1965    011502  045     101     102  MSG2::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.%N/
        011505  125     123     040
        011510  124     111     115
        011513  105     055     117
        011516  125     124     040
        011521  124     122     101
        011524  120     040     103
        011527  101     125     123
        011532  105     104     040
        011535  102     131     040
        011540  127     122     111
        011543  124     105     040
        011546  101     124     124
        011551  105     115     120
        011554  124     056     045
        011557  116     000
1966    011561  045     101     104  MSG3::  .ASCIZ  /%ADHV MAY BE AT THE WRONG Q-BUS ADDRESS.%N%N/
        011564  110     126     040
        011567  115     101     131
        011572  040     102     105
        011575  040     101     124
        011600  040     124     110
        011603  105     040     127
        011606  122     117     116
        011611  107     040     121
        011614  055     102     125
        011617  123     040     101
        011622  104     104     122
        011625  105     123     123
        011630  056     045     116
        011633  045     116     000
1967
1968                                .EVEN
```

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

011636                          BGNMSG  ER0503
011636                                                                  ER0503::

                                PRINTB  #EF0503,R1      ;PRINT THE MESSAGE.
011636  010146                                                                  MOV     R1,-(SP)
011640  012746  004247                                                          MOV     #EF0503,-(SP)
011644  012746  000002                                                          MOV     #2,-(SP)
011650  010600                                                                  MOV     SP,R0
011652  104414                                                                  TRAP    C$PNTB
011654  062706  000006                                                          ADD     #6,SP

                        ENDMSG
011660                                                                  L10003:
011660  104423                                                                  TRAP    C$MSG
```

```
                        .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.
                        ;*************************************************************************
011662                          BGNMSG  ER1603
011662                                                                  ER1603::
011662                          SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
011662  004567  172226                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
011666
011666  010146                  PRINTB #EF0503,R1       ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
011670  012746  004247                                                  MOV     R1,-(SP)
011674  012746  000002                                                  MOV     #EF0503,-(SP)
011700  010600                                                          MOV     #2,-(SP)
011702  104414                                                          MOV     SP,R0
011704  062706  000006                                                  TRAP    C$PNTB
                                                                        ADD     #6,SP
011710  016702  172174          MOV     ERRMSG,R2       ;GET THE "TEST MESSAGE".
011714                          PRINTB #EF1601,R2       ;PRINT "TEST ABORTED" MESSAGE.
011714  010246                                                          MOV     R2,-(SP)
011716  012746  004327                                                  MOV     #EF1601,-(SP)
011722  012746  000002                                                  MOV     #2,-(SP)
011726  010600                                                          MOV     SP,R0
011730  104414                                                          TRAP    C$PNTB
011732  062706  000006                                                  ADD     #6,SP
011736                  PASS            ;RESTORE THE CONTENTS OF THE GPRS.
011736  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
011740                  ENDMSG
011740                                                          L10004:
011740  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER6401 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which prints additional error
                        ;*      information after the error message header.
                        ;*      This subroutine is passed a GPR containing flags which indicate
                        ;*      the line(s) for which the error condition should be reported.
                        ;*
                        ;* INPUTS:      R1 - Address of the message to be printed by this routine.
                        ;*              R5 - Contains the error flags, (1 flag per line).
                        ;*
                        ;* OUTPUTS:     Messages are printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Load the address of the message in R1.
                        ;*                      Include the label "ER6401" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The output format of this message is:
                        ;*                      TEXT MESSAGE
                        ;*                             #nn
                        ;*                             #nn
                        ;*                              ...
                        ;*              Where each "#nn" is the number of a line with the error.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*************************************************************************

011742                          BGNMSG  ER6401
011742                                                                  ER6401::
011742                          SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
011742  004567  172146                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

011746  005002                  CLR     R2              ;CLEAR LINE NUMBER TO ZERO.
011750  012703  000010          MOV     #NUMLNS,R3      ;SET UP MAX LINE COUNT.
011754                          PRINTB  #EF0503,R1      ;PRINT MESSAGE.
011754  010146                                                          MOV     R1,-(SP)
011756  012746  004247                                                  MOV     #EF0503,-(SP)
011762  012746  000002                                                  MOV     #2,-(SP)
011766  010600                                                          MOV     SP,R0
011770  104414                                                          TRAP    C$PNTB
011772  062706  000006                                                  ADD     #6,SP
011776  000241                  CLC                     ;CLEAR CARRY.
012000  006205          2$:     ASR     R5              ;SHIFT FLAG OUT INTO CARRY BIT.
012002  103011                  BCC     4$              ;SKIP ERROR REPORT IF CLEAR.
012004                          PRINTB  #EF6401,R2      ;PRINT MESSAGE.
012004  010246                                                          MOV     R2,-(SP)
012006  012746  004471                                                  MOV     #EF6401,-(SP)
012012  012746  000002                                                  MOV     #2,-(SP)
012016  010600                                                          MOV     SP,R0
012020  104414                                                          TRAP    C$PNTB
012022  062706  000006                                                  ADD     #6,SP
012026  005202          4$:     INC     R2              ;INCREMENT LINE COUNT.
012030  020302                  CMP     R3,R2           ;CHECK IF MAX LINE COUNT EXCEEDED.
012032  001362                  BNE     2$              ;LOOP IF NOT DONE.
012034                  60$:    PASS                    ;RESTORE THE SAVED CONTENTS OF THE GPRS.
012034  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
012036                          ENDMSG
012036                                                                  L10005:
```

GLOBAL ERROR REPORTING ROUTINE - ER6401 -

012036 104423 TRAP C$MSG

GLOBAL ERROR REPORTING ROUTINE - ER7801 -

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER7801 -
                        ;*******************************************************************************
                        ;*      This is an error reporting subroutine which prints an additional error
                        ;*      message whose address is passed as an input parameter.  A line number
                        ;*      is included at the end of the message.
                        ;*
                        ;* INPUTS:      R1 - Address of the message to print.
                        ;*              R3 - Number of line on which error occurred.
                        ;*
                        ;* OUTPUTS:     A messages is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Load the address of the message in R1.
                        ;*                      Load the line number into R3.
                        ;*                      Include the label "ER7801" 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.
                        ;*******************************************************************************

012040                          BGNMSG  ER7801
012040                                                          ER7801::

012040                          PRINTB  #EF7801,R1,R3   ;PRINT THE MESSAGE.
012040  010346                                                          MOV     R3,-(SP)
012042  010146                                                          MOV     R1,-(SP)
012044  012746  004546                                                  MOV     #EF7801,-(SP)
012050  012746  000003                                                  MOV     #3,-(SP)
012054  010600                                                          MOV     SP,R0
012056  104414                                                          TRAP    C$PNTB
012060  062706  000010                                                  ADD     #10,SP

012064                          ENDMSG
012064                                                          L10006:
012064  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER8401 -
                        ;*************************************************************************
                        ;*      This error reporting subroutine is intended to report interactions
                        ;*      which have been found between a modem signal and other modem signals.
                        ;*      It analyzes the modem status which is stored in the STAT storage area
                        ;*      and reports any discrepancies which are found between this stored data
                        ;*      and the present state of the STAT registers.  Specified bits on the
                        ;*      line associated with the specified line are ignored.
                        ;*
                        ;* INPUTS:      R1 - Address of signal name message.
                        ;*              R2 - Bit map of bits to ignore on specified line.
                        ;*              R3 - Number of specified line.
                        ;*              CSRA - Contains the address of the DUT CSR.
                        ;*              NUMLNS - Equated to the number of lines on the DUT.
                        ;*              STATA - Contains the address of the DUT STAT register.
                        ;*              STSTB - Label at base of STAT storage table.
                        ;*              TXRLNB - Label at base of TX/RX line number association table.
                        ;*
                        ;* OUTPUTS:     A messages is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER8401" 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.
                        ;*************************************************************************

012066                          BGNMSG  ER8401
012066                                                                  ER8401::
012066                          SAVE                    ;PRESERVE THE CONTENTS OF THE GPRS.
012066  004567  172022                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

012072                          PRINTB  #EF8401,R1,R3   ;PRINT THE BASIC MESSAGE.
012072  010346                                                          MOV     R3,-(SP)
012074  010146                                                          MOV     R1,-(SP)
012076  012746  004604                                                  MOV     #EF8401,-(SP)
012102  012746  000003                                                  MOV     #3,-(SP)
012106  010600                                                          MOV     SP,R0
012110  104414                                                          TRAP    C$PNTB
012112  062706  000010                                                  ADD     #10,SP

012116  010167  000204          MOV     R1,44$          ;SAVE THE ADDRESS OF THE SIGNAL NAME MESSAGE.
012122  005001                  CLR     R1              ;CLEAR THE LINE COUNTER.
012124  012704  002644          MOV     #STSTB,R4       ;SET UP STAT STORAGE POINTER TO BASE OF TABLE.
012130  010177  170106  2$:     MOV     R1,@CSRA        ;SET UP THE CSR IND.ADR.REG FIELD.
012134  017700  170110          MOV     @STATA,R0       ;GET THE CONTENTS OF THIS LINE'S STAT REGISTER.
012140  011405                  MOV     (R4),R5         ;GET THE PREVIOUS CONTENTS FROM STORAGE.
012142  040005                  BIC     R0,R5
012144  042400                  BIC     (R4)+,R0
012146  050005                  BIS     R0,R5           ;XOR PRESENT AND STORED STAT VALUES.
012150  012700  043777          MOV     #43777,R0       ;PREPARE TO MASK OUT UNUSED BITS.
012154  120163  004044          CMPB    R1,TXRLNB(R3)   ;IS THIS LINE ASSOCIATED WITH SPECIFIED LINE?
012160  001002                 BNE     4$              ;DON'T MASK OUT SPECIFIED BITS IF IT IS NOT.
012162  056600  000006          BIS     R2SLOT(SP),R0   ;MASK OUT SPECIFIED BITS.
012166  040005          4$:     BIC     R0,R5           ;GET BIT MAP OF UNDESIRED CHANGES.
012170  032705  100000          BIT     #BIT15,R5       ;CHECK FOR DSR SIGNAL INTERACTION.
```

GLOBAL ERROR REPORTING ROUTINE - ER8401 -

```
2143 012174  001404                    BEQ     6$              ;SKIP PRINTING LINE IF NO DSR INTERACTION.
2144 012176  012702  010304             MOV     #EM8403,R2      ;SELECT DSR ERROR MESSAGE.
2145 012202  004767  000064            JSR     PC,40$          ;PRINT THE LINE OF THE ERROR MESSAGE.
2146 012206  032705  020000  6$:       BIT     #BIT13,R5       ;CHECK FOR RI SIGNAL INTERACTION.
2147 012212  001404                    BEQ     8$              ;SKIP PRINTING LINE IF NO RI INTERACTION.
2148 012214  012702  010310             MOV     #EM8404,R2      ;SELECT RI ERROR MESSAGE.
2149 012220  004767  000046            JSR     PC,40$          ;PRINT THE LINE OF THE ERROR MESSAGE.
2150 012224  032705  010000  8$:       BIT     #BIT12,R5       ;CHECK FOR DCD SIGNAL INTERACTION.
2151 012230  001404                    BEQ     10$             ;SKIP PRINTING LINE IF NO DCD INTERACTION.
2152 012232  012702  010313             MOV     #EM8405,R2      ;SELECT DCD ERROR MESSAGE.
2153 012236  004767  000030            JSR     PC,40$          ;PRINT THE LINE OF THE ERROR MESSAGE.
2154 012242  032705  004000  10$:       BIT     #BIT11,R5       ;CHECK FOR CTS SIGNAL INTERACTION.
2155 012246  001404                    BEQ     12$             ;SKIP PRINTING LINE IF NO CTS INTERACTION.
2156 012250  012702  010317             MOV     #EM8406,R2      ;SELECT CTS ERROR MESSAGE.
2157 012254  004767  000012            JSR     PC,40$          ;PRINT THE LINE OF THE ERROR MESSAGE.
2158
2159 012260  005201  12$:       INC     R1              ;SELECT NEXT LINE.
2160 012262  020127  000010             CMP     R1,#NUMLNS      ;ALL LINES DONE?
2161 012266  002720                    BLT     2$              ;LOOP IF NOT ALL LINES DONE.
2162 012270  000417                    BR      60$             ;EXIT THIS ROUTINE.
2163                                   ;+
2164                                   ; Local error message line printing routine.
2165                                   ;-
2166 012272  40$:       PRINTX  #EF8402,44$,R3,R2,R1
     012272  010146                                                          MOV     R1,-(SP)
     012274  010246                                                          MOV     R2,-(SP)
     012276  010346                                                          MOV     R3,-(SP)
     012300  016746  000022                                                  MOV     44$,-(SP)
     012304  012746  004676                                                  MOV     #EF8402,-(SP)
     012310  012746  000005                                                  MOV     #5,-(SP)
     012314  010600                                                          MOV     SP,R0
     012316  104415                                                          TRAP    C$PNTX
     012320  062706  000014                                                  ADD     #14,SP
2167 012324  000207                    RTS     PC              ;
2168 012326  000000  44$:       .WORD   0               ;LOCAL STORAGE FOR ADDRESS OF SIGNAL NAME.
2169 012330          60$:       PASS                    ;RESTORE ALL THE GPRS TO THE PRESERVED VALUES.
     012330  004736                            JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2170 012332                    ENDMSG
     012332                                                          L10007:
     012332  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9001 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which reports an unexpected
                        ;*      code which has been found in the DUT CSR.  This code can be a BMP
                        ;*      code, a Self-test code, or a Modem Status code.
                        ;*
                        ;* INPUTS:      R1 - Address of message to print first.
                        ;*              R2 - Single byte code which has been read from the DUT.
                        ;*              R4 - Line number associated with the code.
                        ;*
                        ;* OUTPUTS:     A messages is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER9001" 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.
                        ;*************************************************************************

012334                          BGNMSG  ER9001
012334                                                          ER9001::
                        
012334                          PRINTB  #EF9001,R1      ;REPORT TYPE OF CODE FOUND.
012334  010146                                                          MOV     R1,-(SP)
012336  012746  005013                                                  MOV     #EF9001,-(SP)
012342  012746  000002                                                  MOV     #2,-(SP)
012346  010600                                                          MOV     SP,R0
012350  104414                                                          TRAP    C$PNTB
012352  062706  000006                                                  ADD     #6,SP
012356                          PRINTX  #EF9002,R4      ;REPORT THE LINE NUMBER OF THE CODE.
012356  010446                                                          MOV     R4,-(SP)
012360  012746  005075                                                  MOV     #EF9002,-(SP)
012364  012746  000002                                                  MOV     #2,-(SP)
012370  010600                                                          MOV     SP,R0
012372  104415                                                          TRAP    C$PNTX
012374  062706  000006                                                  ADD     #6,SP
012400                          PRINTX  #EF9003,R2      ;REPORT THE CODE WHICH WAS FOUND.
012400  010246                                                          MOV     R2,-(SP)
012402  012746  005147                                                  MOV     #EF9003,-(SP)
012406  012746  000002                                                  MOV     #2,-(SP)
012412  010600                                                          MOV     SP,R0
012414  104415                                                          TRAP    C$PNTX
012416  062706  000006                                                  ADD     #6,SP

012422                          ENDMSG
012422                                                  L10010:
012422  104423                                                          TRAP    C$MSG
```

GLOBAL ERROR REPORTING ROUTINE - ER9002 -

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9002 -
                        ;*******************************************************************************
                        ;*      This is an error reporting subroutine which is intended for use in the
                        ;*      tranmsission and reception tests.  It reports the type of error which
                        ;*      has occurred when incorrect data is received from the DUT.  This
                        ;*      routine also reports the read and expected data values.
                        ;*
                        ;* INPUTS:      R1 - Address of message to print first.
                        ;*              R2 - Data byte read from the DUT.
                        ;*              R3 - Line number multiplied by 2.
                        ;*              R4 - Expected data byte, bit 15 set if "NONE".
                        ;*
                        ;* OUTPUTS:     A messages is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER9002" 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: PRTLPR.
                        ;*******************************************************************************

012424                          BGNMSG  ER9002
012424                                                          ER9002::
012424  006203                   ASR    R3              ;CALCULATE THE LINE NUMBER.
012426  042702  177400           BIC    #177400,R2      ;MASK OUT ALL BUT DATA IN READ CHAR.
012432                          PRINTB  #EF9006,R1,R3   ;PRINT THE FIRST LINE OF THE MESSAGE.
012432  010346                                                  MOV     R3,-(SP)
012434  010146                                                  MOV     R1,-(SP)
012436  012746  005257                                         MOV     #EF9006,-(SP)
012442  012746  000003                                         MOV     #3,-(SP)
012446  010600                                                  MOV     SP,R0
012450  104414                                                  TRAP    C$PNTB
012452  062706  000010                                         ADD     #10,SP
012456                          PRINTX  #EF9004,#EM9010,R2      ;PRINT ACTUAL DATA.
012456  010246                                                  MOV     R2,-(SP)
012460  012746  010672                                         MOV     #EM9010,-(SP)
012464  012746  005176                                         MOV     #EF9004,-(SP)
012470  012746  000003                                         MOV     #3,-(SP)
012474  010600                                                  MOV     SP,R0
012476  104415                                                  TRAP    C$PNTX
012500  062706  000010                                         ADD     #10,SP
012504  005704                   TST    R4              ;CHECK FOR "NONE" CODE SET IN EXPECTED DATA.
012506  100414                  BMI     2$              ;BRANCH TO PRINT "NONE" MESSAGE IF FLAG SET.
012510                          PRINTX  #EF9004,#EM9009,R4      ;PRINT EXPECTED DATA.
012510  010446                                                  MOV     R4,-(SP)
012512  012746  010646                                         MOV     #EM9009,-(SP)
012516  012746  005176                                         MOV     #EF9004,-(SP)
012522  012746  000003                                         MOV     #3,-(SP)
012526  010600                                                  MOV     SP,R0
012530  104415                                                  TRAP    C$PNTX
012532  062706  000010                                         ADD     #10,SP
012536  000412                  BR      60$             ;EXIT THIS ROUTINE.
012540                  2$:     PRINTX  #EF9005,#EM9009 ;PRINT MESSAGE INDICATING NO EXPECTED DATA.
012540  012746  010646                                         MOV     #EM9009,-(SP)
012544  012746  005226                                         MOV     #EF9005,-(SP)
```

GLOBAL ERROR REPORTING ROUTINE - ER9002 -

```
        012550  012746  000002                                                  MOV     #2,-(SP)
        012554  010600                                                          MOV     SP,R0
        012556  104415                                                          TRAP    C$PNTX
        012560  062706  000006                                                  ADD     #6,SP
  2233  012564  004767  002162          60$:    JSR     PC,PRTLPR       ;PRINT CONTENTS OF THE LPR REGISTER.
  2234  012570                                  ENDMSG
        012570                                                          L10011:
        012570  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.
                                ;*******************************************************************************

                                        BGNMSG  ER9101
                                                                                ER9101::

                                        PRINTB  #EF9006,R2,R1   ;REPORT THE STRING FOLLOWED BY THE NUMBER.
012572  010146                                                          MOV     R1,-(SP)
012574  010246                                                          MOV     R2,-(SP)
012576  012746  005257                                                  MOV     #EF9006,-(SP)
012602  012746  000003                                                  MOV     #3,-(SP)
012606  010600                                                          MOV     SP,R0
012610  104414                                                          TRAP    C$PNTB
012612  062706  000010                                                  ADD     #10,SP

                                ENDMSG
012616                                                  L10012:
012616  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9301 -
                        ;*******************************************************************************
                        ;*      This is an error reporting subroutine which prints any BMP codes
                        ;*      that are found in the BMP code queue, together with the the number of
                        ;*      the test that was executing at the time the BMP code was logged.
                        ;*
                        ;* INPUTS:      R1 - The address of the first message to be reported.
                        ;*              R2 - The address of the next empty cell in the queue.
                        ;*
                        ;* OUTPUTS:     The test number followed by the BMP code are printed at the
                        ;*              operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER9301" 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.
                        ;*******************************************************************************

012620                          BGNMSG  ER9301
012620                                                                  ER9301::
012620                          SAVE                    ;SAVE THE GPRS ON THE STACK.
012620  004567  171270                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

012624                          PRINTB  #EF0503,R1      ;REPORT UNEXPECTED BMP CODES FOUND.
012624  010146                                                  MOV     R1,-(SP)
012626  012746  004247                                          MOV     #EF0503,-(SP)
012632  012746  000002                                          MOV     #2,-(SP)
012636  010600                                                  MOV     SP,R0
012640  104414                                                  TRAP    C$PNTB
012642  062706  000006                                          ADD     #6,SP
012646  012703  002444                MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
012652  012705  011147                MOV     #EM9302,R5      ;GET THE MESSAGE TO BE REPORTED.
012656  012301        2$:             MOV     (R3)+,R1        ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
012660  012304                        MOV     (R3)+,R4        ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
012662  004767  000056                JSR     PC,50$          ;GO REPORT THE BMP CODE.
012666  020302                        CMP     R3,R2           ;CHECK IF ALL CODES HAVE BEEN REPORTED.
012670  103772                        BLO     2$              ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
                        ;+
                        ; Check if overflow has occurred.
                        ;  The conditions for overflow are: the pointer contains the address of the
                        ;  last cell in the queue, and a bmp code has already been written into that
                        ;  cell.
                        ;-
012672  020227  002640                CMP     R2,#BMPCQE-4    ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
012676  001036                        BNE     60$             ;EXIT IF NOT AT THE LAST LOCATION.
012700  005762  000002                TST     2(R2)           ;CHECK FOR A BMP CODE IN THE LAST CELL
012704  001433                      BEQ     60$             ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
012706  012301                        MOV     (R3)+,R1        ;GET THE TEST NUMBER OFF THE QUEUE.
012710  011304                        MOV     (R3),R4         ;GET THE BMP CODE OFF THE QUEUE.
012712  012705  011177                MOV     #EM9303,R5      ;SELECT THE MESSAGE TO BE REPORTED.
012716                              PRINTX  #EF9302         ;REPORT OVERFLOW CONDITION.
012716  012746  005373                                                  MOV     #EF9302,-(SP)
012722  012746  000001                                                  MOV     #1,-(SP)
012726  010600                                                          MOV     SP,R0
012730  104415                                                          TRAP    C$PNTX
```

GLOBAL ERROR REPORTING ROUTINE - ER9301 -

```
      012732  062706  000004                                                    ADD     #4,SP
2306  012736  004767  000002            JSR     PC,50$          ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
2307  012742  000414                    BR      60$             ;EXIT.
2308
2309  012744                    50$:    PRINTX  #EF9301,R5,R1,R4  ;PRINT THE MESSAGE.
      012744  010446                                                            MOV     R4,-(SP)
      012746  010146                                                            MOV     R1,-(SP)
      012750  010546                                                            MOV     R5,-(SP)
      012752  012746  005315                                                    MOV     #EF9301,-(SP)
      012756  012746  000004                                                    MOV     #4,-(SP)
      012762  010600                                                            MOV     SP,R0
      012764  104415                                                            TRAP    C$PNTX
      012766  062706  000012                                                    ADD     #12,SP
2310  012772  000207                    RTS     PC              ;RETURN.
2311  012774                    60$:    PASS                    ;RESTORE THE GPR CONTENTS.
      012774  004736                            JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2312
2313  012776                    ENDMSG
      012776                                                            L10013:
      012776  104423                                                            TRAP    C$MSG
```

GLOBAL SUBROUTINES SECTION

```
                        .SBTTL  GLOBAL SUBROUTINES SECTION
                        ;**********************************************************************
                        ;
                        ;               SKL3.P11
                        ;
                        ;**********************************************************************


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

```
                                .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.
                                ;-- ************************************************************************
                                ALTFLD:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
013000  004567  171110                          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.
                                ;-
013004  010400                      MOV     R4,R0               ;CALCULATE THE NEW CONTENTS OF THE
013006  005100                      COM     R0                  ;  REGISTER FIELDS WHICH ARE TO BE
013010  040002                      BIC     R0,R2               ;  ALTERRED BY THIS ROUTINE.
013012  016705  167252              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).
                                ;-
013016  000241                      CLC                         ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
013020  006003                  2$: ROR     R3                  ;GET THE LINE SELECT BIT FOR THIS LINE.
013022  103006                      BCC     4$                  ;SKIP SETUP IF LINE IS NOT SELECTED.
013024  010577  167212              MOV     R5,@CSRA            ;SET DUT CSR IND.ADR.REG FIELD TO THIS LINE.
013030  011100                      MOV     (R1),R0             ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
013032  040400                      BIC     R4,R0               ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
013034  050200                      BIS     R2,R0               ;OR IN THE NEW STATES OF THE FIELDS.
013036  010011                      MOV     R0,(R1)             ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
013040  005205                  4$: INC     R5                  ;SET LINE NUMBER TO THE NEXT LINE.
013042  005703                      TST     R3                  ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
013044  001365                      BNE     2$                  ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.
```

GLOBAL SUBROUTINE - ALTFLD -

```
   2386
   2387 013046                          60$:    PASS                    ;RESTORE GPRS.
        013046  004736                                    JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
   2388 013050  000207                          RTS     PC              ;RETURN TO CALLING ROUTNE.
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - ASLNTL -
                                ;** ************************************************************************
                                ;*              - Setup Associated Line Number Tables Routine -
                                ;*      This routine sets up the two tables which are contain information
                                ;*      about the TX/RX line which is associated with a particular RX/TX
                                ;*      line.  One table is a table of words which contains word offset
                                ;*      values and the other table is a table of bytes which contains
                                ;*      line number values.
                                ;*
                                ;* INPUTS:      LOPBCK - Storage for the type of loopback on the DUT.
                                ;*              NUMLNS - Equated to the number of lines on the DUT.
                                ;*              STGTRB - Label at base of staggered line association tbl.
                                ;*              TXRLNB - Label at base of byte TX/RX line number table.
                                ;*              TXRXLB - Label at base of word TX/RX line number table.
                                ;*              TXRXLE - Label at end of word TX/RX line number table.
                                ;*
                                ;* OUTPUTS:     TXRXL, TXRLN - Tables initialized for selected loopback.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,ASLNTL
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;-- ************************************************************************

013052                          ASLNTL:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
013052  004567  171036                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
013056  126727  167154  000002          CMPB    LOPBCK,#2       ;TEST FOR STAGGERED LOOPBACK.
013064  001411                        BEQ     4$              ;GO SET UP STAGGERED TABLE IF STAGGERED LPBCK.
                                ;+
                                ; Set up the word table for non-staggered loopback.
                                ;-
013066  005005                          CLR     R5              ;CLEAR THE LINE COUNTER
013070  010565  004004          2$:     MOV     R5,TXRXLB(R5)   ;SET UP A WORD OF THE TABLE.
013074  005205                          INC     R5
013076  005205                          INC     R5              ;SET LINE COUNTER TO NEXT LINE OFFSET.
013100  020527  000020                  CMP     R5,#2*NUMLNS    ;TEST FOR ALL LINES DONE.
013104  002771                        BLT     2$              ;LOOP UNTIL ALL LINES DONE.
013106  000411                        BR      8$              ;GO SET UP THE BYTE TABLE.
                                ;+
                                ; Set up the word table for staggered loopback.
                                ;-
013110  012701  004064          4$:     MOV     #STGTRB,R1      ;SET UP THE SOURCE POINTER.
013114  012702  004004                  MOV     #TXRXLB,R2      ;SET UP THE DESTINATION POINTER.
013120  112122                  6$:     MOVB    (R1)+,(R2)+     ;MOVE A BYTE INTO THE TABLE.
013122  105022                          CLRB    (R2)+           ;CLEAR THE UPPER BYTE OF WORD TABLE ENTRY.
013124  020227  004044                  CMP     R2,#TXRXLE      ;COMPARE POINTER WITH END ADR OF TABLE.
013130  002773                        BLT     6$              ;LOOP IF NOT AT END YET.
                                ;+
                                ; Set up the byte table based on the word association table.
                                ;-
013132  012701  004004          8$:     MOV     #TXRXLB,R1      ;SET UP THE SOURCE POINTER.
013136  012702  004044                  MOV     #TXRLNB,R2      ;SET UP THE DESTINATION POINTER.
013142  012103                  10$:    MOV     (R1)+,R3        ;GET THE WORD OFFSET VALUE FROM WORD TABLE.
013144  006203                          ASR     R3              ;DIVIDE BY 2 TO GET LINE NUMBER VALUE.
013146  110322                          MOVB    R3,(R2)+        ;LOAD THE BYTE LINE NUMBER INTO TABLE.
013150  020127  004044                  CMP     R1,#TXRXLE      ;COMPARE SOURCE POINTER WITH ADR OF TABLE END.
```

GLOBAL SUBROUTINE - ASLNTL -

```
2446 013154  002772                    BLT     10$             ;LOOP IF NOT AT END OF TABLE YET.
2447
2448 013156                      60$:   PASS                    ;RESTORE GPRS.
     013156  004736                            JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
2449 013160  000207                    RTS     PC
```

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.
                                ;-- ************************************************************************

013162                          CALMSL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
013162  004567  170726                          JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
013166  005067  000210                  CLR     62$                     ;CLEAR THE 2ND TIME FLAG.
                                ;+
                                ; Synchronize with the LTC.
                                ;-
013172  012705  000001          2$:     MOV     #1,R5                   ;SET OUTER LOOP COUNTER TO 1 LOOP.
                                                                        ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE <**
                                                                        ;FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <**
013176  005000                          CLR     R0                      ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
013200  012767  000001  167120          MOV     #1,TIMER1               ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
013206  005767  167114          4$:     TST     TIMER1                  ;CHECK FOR COUNTER HAVING GONE TO ZERO.
013212  001410                          BEQ     6$                      ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
013214  005200                          INC     R0                      ;COUNT THIS ITERATION OF THE INNER LOOP.
013216  001373                          BNE     4$                      ;LOOP IF COUNTER HAS NOT TURNED OVER.
013220  005305                          DEC     R5                      ;DECREMENT THE INNER LOOP COUNTER.
013222  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.
                                    ;-
013224  005067  167074                  CLR     CLKHRZ                  ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
013230  000241                          CLC                             ;INDICATE FAILURE FOR RETURN.
013232  000461                          BR      60$                     ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
                                ;+
                                ; We are now synchronized with the LTC.
                                ; Set up for the calibration loop.
                                ;-
013234  012704  002326          6$:     MOV     #TIMER1,R4              ;WILL TEST TIMER1 IN THE LOOP BELOW.
```

GLOBAL SUBROUTINE - CALMSL -

```
2507 013240  005001                    CLR     R1              ;CLEAR THE OUTER LOOP COUNTER.
2508 013242  005002                    CLR     R2              ;INDICATE TO CHECK ALL BITS OF TIMER1.
2509 013244  005003                    CLR     R3              ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2510 013246  012714  000001            MOV     #1,(R4)         ;LOAD TIMER1 WITH COUNT OF 1.
2511
2512 013252  016705  167062    8$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2513 013256  011400            10$:    MOV     (R4),R0         ;GET THE TIMER1 VALUE.
2514 013260  010067  000120            MOV     R0,64$          ;SAVE WORD (LIKE IN THE REAL LOOP).
2515 013264  040200                    BIC     R2,R0           ;LEAVE ALL THE BITS.
2516 013266  020003                    CMP     R0,R3           ;COMPARE AGAINST ZERO.
2517 013270  000261                    SEC                     ;SET CARRY IN CASE OF SUCCESS.
2518 013272  001406                    BEQ     12$             ;EXIT LOOP IF TIMER1 HAS CLEARED.
2519 013274  005305                    DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2520 013276  001367                    BNE     10$             ;LOOP IF MS NOT UP.
2521 013300  005301                    DEC     R1              ;DECREMENT THE MS TIME COUNT.
2522 013302  001363                    BNE     8$              ;KEEP LOOPING.
2523 013304  004767  001214            JSR     PC,OOPS         ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
2524                            ;+
2525                            ; We have now have loop count information for one clock tick.
2526                            ; We have negative of number of outer loops in R1, each is MSLCNT inner loops.
2527                            ; We have the portion of the last outer loop not executed, in R5.
2528                            ; Now we calculate the total number of inner loops executed.
2529                            ;-
2530 013310  005401    12$:     NEG     R1              ;GET NUMBER OF OUTER LOOPS.
2531 013312  016702  167022            MOV     MSLCNT,R2       ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2532 013316  010203                    MOV     R2,R3           ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2533 013320  160502                    SUB     R5,R2           ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2534 013322  010204                    MOV     R2,R4           ;   AND ADD TO ACCUMULATOR LSWORD.
2535 013324  005005                    CLR     R5              ;CLEAR ACCUMULATOR MSWORD.
2536 013326  005301    14$:     DEC     R1              ;CHECK R1 FOR 0 CONDITION
2537 013330  100403                    BMI     16$             ; SKIP MULTIPLICATION IF ZERO
2538 013332  060304                    ADD     R3,R4           ;MULTIPLY NUMBER OF INNER
2539 013334  005505                    ADC     R5              ; LOOPS PER OUTER LOOP BY
2540 013336  000773                    BR      14$             ;NUMBER OF OUTER LOOPS PERFORMED.
2541                            ;+
2542                            ; Divide the total number of inner loops by the number of MS per LTC tick.
2543                            ;-
2544 013340  016701  166772    16$:     MOV     MSTICK,R1       ;# OF MS PER LTC TICK IS DIVISOR.
2545 013344  010403                    MOV     R4,R3           ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2546 013346  010502                    MOV     R5,R2           ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2547 013350  004767  003012            JSR     PC,UNSDIV       ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
2548 013354  103402                    BCS     18$             ;BYPASS OOPS IF WE'RE OK.
2549 013356  004767  001142            JSR     PC,OOPS         ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2550 013362  010167  166752    18$:     MOV     R1,MSLCNT       ;SET NEW VALUE FOR MS LOOP COUNT.
2551 013366  005167  000010            COM     62$             ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2552 013372  001277                    BNE     2$              ;BRANCH IF ONLY ONE ITERATION DONE.
2553 013374  000261                    SEC                     ;SET THE SUCCESS FLAG FOR EXIT.
2554
2555 013376            60$:     PASS                            ;RESTORE GPRS,
     013376  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
2556 013400  000207                    RTS     PC              ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
2557
2558 013402  000000    62$:     .WORD   0               ;2ND CALIBRATION ITERATION FLAGS.
2559 013404  000000    64$:     .WORD   0               ;DUMMY WORD FOR STORAGE OF THE READ WORD.
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - CHKBMP -
                                ;** ************************************************************************
                                ;*                          - CHECK IF CHARACTER IS A BMP CODE -
                                ;*      This subroutine is used to check for BMP codes.
                                ;*      If a BMP code is detected, it will be saved on the queue to be reported
                                ;*      later.  The carry is used as a flag to indicate a code has been found.
                                ;*
                                ;* INPUTS:      R2 - Contains the data to be checked.
                                ;*
                                ;* OUTPUTS:     R1 - Contians the message to be reported.
                                ;*              ERRBLK - Contains the error reporting routine.
                                ;*              Carry bit is used to indicate a BMP code found, Carry set.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,CHKBMP
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
                                ;-- ************************************************************************

013406                          CHKBMP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
013406  004567  170502                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
013412  012700  170301                 MOV     #170301,R0              ;SET UP THE FLAGS OF A BMP CODE.
013416  040200                         BIC     R2,R0                   ;TRY TO CLEAR THE BMP CODE FLAGS.
013420  001011                        BNE     2$                      ;IF NOT A BMP CODE, EXIT WITH FAILURE.
013422  004767  002116                JSR     PC,SAVBMP               ;SAVE THE BMP CODE ON THE QUEUE.
013426  012701  006634                 MOV     #EM5303,R1             ;PASS THE MESSAGE TO BE REPORTED.
013432  012767  011662  170452         MOV     #ER1603,ERRBLK         ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
013440  000261                         SEC                             ;PASS FLAG TO INDICATE SUCCESS, BMP CODE FOUND.
013442  000401                        BR      60$                     ;EXIT.
013444  000241                2$:      CLC                             ;PASS FLAG TO INDICATE FAILURE.
013446                          60$:    PASS    R1                      ;RESTORE GPRS, EXCEPT
013446  010166  000004                                  MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
013452  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                ;R1 - CONTAINS THE ADDRESS OF ERROR MESSAGE.
                                ;CARRY BIT - SET INDICATES SUCCESS.
013454  000207                        RTS     PC
```

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

013456                  CKTRAP:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
013456  004567  170432                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
013462  005067  166624          CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAGS.
013466  011011                  MOV     (R0),(R1)       ;PERFORM THE MOVE IN QUESTION.
013470  005767  166616  ADRPTR:: TST     TP4FLG          ;CHECK FOR OCCURENCE OF TRAP.
013474  000261                  SEC                     ;INDICATE SUCCESS.
013476  001401                  BEQ     60$             ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
013500  000241                  CLC                     ;INDICATE FAILURE.
013502                  60$:    PASS                    ;RESTORE GPRS.
013502  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
013504  000207                  RTS     PC
```

GLOBAL SUBROUTINE - CLNRST -

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

013506                  CLNRST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
013506  004567  170402                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; Reset the DUT.
                        ;  This routine reports errors with numbers from ERRNBR thru ERRNBR+2.
                        ;-
013512  004767  001654          JSR     PC,RESETT       ;RESET THE DUT TO A KNOWN STATE.
013516  103002                  BCC     60$             ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
                        ;+
                        ; Purge the FIFO of error codes, save any BMP codes found.
                        ;-
013520  004767  001310          JSR     PC,PUFIFO       ;PURGE THE FIFO.

013524                  60$:                            ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
013524                  PASS                            ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
013524  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                        ;CARRY BIT:IF CLEAR, THEN ABORT THE TEST.
013526  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.
                                ;-- ************************************************************************

013530                          CLR16W:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
013530  004567  170360                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
013534  012701  000020                  MOV     #16.,R1                 ;SET THE LOOP COUNTER TO 16.
013540  005020                  2$:     CLR     (R0)+                   ;CLEAR A WORD OF MEMORY.
013542  005301                          DEC     R1                      ;COUNT THIS LOOP.
013544  001375                          BNE     2$                      ;LOOP IF NOT 16 WORD CLEARED.
013546                          60$:    PASS                            ;RESTORE GPRS.
013546  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
013550  000207                          RTS     PC
```

GLOBAL SUBROUTINE - CMPMST -

```
                        .SBTTL  GLOBAL SUBROUTINE                - CMPMST -
                        ;++ ************************************************************************
                        ;*                        - Compare Modem Status Routine -
                        ;*      This routine is used to compare the present modem status against the
                        ;*      modem status which is stored in the modem status storage table.  It
                        ;*      ignores the states of the specified signals on a specified line.
                        ;*
                        ;* INPUTS:      R1 - Line number of specified line.
                        ;*              R2 - Bit map of bits to ignore on specified line.
                        ;*              CSRA - Contains the address of the DUT CSR.
                        ;*              NUMLNS - Equated to the number of lines on the DUT.
                        ;*              STATA - Contains the address of the DUT STAT register.
                        ;*              STSTB - Label at base of STAT storage table.
                        ;*              TXRLNB - Label at base of TX/RX line number association table.
                        ;*
                        ;* OUTPUTS:     CARRY - Success flag (Set if no discrepancies were found).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CMPMST
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ************************************************************************

013552                  CMPMST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
013552  004567  170336                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
013556  005003                  CLR     R3                      ;CLEAR THE LINE COUNTER.
013560  012704  002644          MOV     #STSTB,R4               ;SET UP STAT STORAGE POINTER TO BASE OF TABLE.
013564  010377  166452  2$:     MOV     R3,@CSRA                ;SET UP THE CSR IND.ADR.REG FIELD.
013570  017700  166454          MOV     @STATA,R0               ;GET THE CONTENTS OF THIS LINE'S STAT REGISTER.
013574  011405                  MOV     (R4),R5                 ;GET THE PREVIOUS CONTENTS FROM STORAGE.
013576  040005                  BIC     R0,R5
013600  042400                  BIC     (R4)+,R0
013602  050005                  BIS     R0,R5                   ;XOR PRESENT AND STORED STAT VALUES.
013604  012700  043777          MOV     #43777,R0               ;PREPARE TO MASK OUT UNUSED BITS.
013610  120301                  CMPB    R3,R1                   ;TEST FOR THIS BEING SPECIFIED LINE.
013612  001001                  BNE     10$                     ;DON'T MASK OUT SPECIFIED BITS IF IT IS NOT.
013614  050200                  BIS     R2,R0                   ;MASK OUT SPECIFIED BITS.
013616  040005          10$:    BIC     R0,R5                   ;GET BIT MAP OF UNDESIRED CHANGES.
013620  001006                  BNE     50$                     ;EXIT WITH FAILURE IF CHANGES OCCURRED.
013622  005203                  INC     R3                      ;SELECT NEXT LINE.
013624  020327  000010          CMP     R3,#NUMLNS              ;ALL LINES DONE?
013630  002755                  BLT     2$                      ;LOOP IF NOT ALL LINES DONE.
013632  000261                  SEC                             ;INDICATE SUCCESS.
013634  000401                  BR      60$                     ;EXIT THIS ROUTINE WITH SUCCESS.

013636  000241          50$:    CLC                             ;INDICATE FAILURE.

013640                  60$:    PASS                            ;RESTORE GPRS.
013640  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
013642  000207                  RTS     PC                      ;  CARRY - SUCCESS FLAG (SET IF SUCCESS).
```

GLOBAL SUBROUTINE - CONMAP -

```
                                .SBTTL  GLOBAL SUBROUTINE                - CONMAP -
                                ;++ ************************************************************************
                                ;*                      - Convert Line bit map.
                                ;*      This subrouitne is used to convert a bit map passed to it , into
                                ;*      another line bit map that is based upon the associated TX/RX line
                                ;*      number/offset table.
                                ;*
                                ;* INPUTS:      R5 - Contains the line bit map to be transformed.
                                ;*              TXRXLB - Base address of associated TX/RX line number table.
                                ;*
                                ;* OUTPUTS:     R5 - Contains an assosciated line bit map.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,CONMAP
                                ;*
                                ;* COMMENTS:    The TX/RX association table must be initialised before this
                                ;*              routine is called.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

013644                          CONMAP::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
013644  004567  170244                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
013650  012702  004004                MOV     #TXRXLB,R2      ;GET THE BASE ADDRESS OF THE LINE ASSOC TABLE.
013654  010503                      MOV     R5,R3           ;COPY THE BIT MAP TO BE TRANSFORMED.
013656  012704  000010              MOV     #NUMLNS,R4      ;SET MAX LINE COUNTER.
013662  005005                      CLR     R5              ;CLEAR ASSOCIATED LINE BIT MAP.
013664  006203                2$:   ASR     R3              ;SHIFT ACTLNS BIT MAP INT BOOLEAN REGISTER.
013666  103005                    BCC     4$              ;SKIP SETTING ASSOCIATED LINE NUMBER BIT MAP.
013670  011201                      MOV     (R2),R1         ;GET ASSOCIATED LINE NUMBER OFFSET FROM TABLE.
013672  006201                      ASR     R1              ;SHIFT RIGHT TO GET LINE NUMB FROM OFFSET.
013674  004767  000414              JSR     PC,LINBIT       ;GENERATE AN SINGLE BIT MAP FOR THIS LINE.
013700  050005                      BIS     R0,R5           ;SET BIT FOR THIS LINE IN ASSOCIATED BIT MAP.
013702  005722                4$:   TST     (R2)+           ;INCREMENT ADDRESS FOR THE NEXT LINE NUMBER.
013704  005304                      DEC     R4              ;DECREMENT LINE COUNT.
013706  001366                    BNE     2$              ;LOOP IF NOT DONE.
013710                          60$:  PASS    R5              ;RESTORE GPRS, EXCEPT
013710  010566  000014                              MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
013714  004736                                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                ;R5 - CONTAINS THE ASSOCIATED LINE BIT MAP.
013716  000207                    RTS     PC
```

GLOBAL SUBROUTINE - DELAY -

```
                                .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.
                                ;*****************************************************************************
013720                          DELAY:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
013720  004567  170170                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
013724  010401                          MOV     R4,R1                   ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
013726  012702  177777                  MOV     #-1,R2                  ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
013732  005003                          CLR     R3                      ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
013734  012704  013756                  MOV     #62$,R4                 ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
013740  004767  000544                  JSR     PC,MSLOOP               ;DELAY THE REQUESTED # OF MS.
013744  103002                          BCC     60$                     ;EXIT ROUTINE IF WE TIMED-OUT.]
013746  004767  000552                  JSR     PC,OOPS                 ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
013752                          60$:    PASS                            ;RESTORE GPRS.
013752  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
013754  000207                          RTS     PC
                                
013756  177777                  62$:    .WORD   -1                      ;DUMMY, NON-ZERO WORD.
```

GLOBAL SUBROUTINE - DODMA -

```
                        .SBTTL  GLOBAL SUBROUTINE               - DODMA -
                        ;** ************************************************************************
                        ;*                      - Initiate DMA Transmission Routine -
                        ;*      This routine writes the DMA parameter to the specified device and
                        ;*      initiates the DMA transmission.
                        ;*
                        ;* INPUTS:      R1 - Line number on which to initiate the DMA.
                        ;*              R2 - Start address of the DMA buffer (16 bit virtual).
                        ;*              R3 - Character count of the DMA buffer.
                        ;*              CSRA - Contains address of the DUT CSR.
                        ;*              IESTAT - Storage for states of the interrupt enable bits.
                        ;*              MMENAB - Memmory management flag (0 if MEM MGT not enabled).
                        ;*              HOST MEM MGT PAR REGISTERS - If MEM MGT is in use.
                        ;*              TXAD1A - Contains address of DMA TX buffer address reg #1.
                        ;*              TXAD2A - Contains address of DMA TX buffer address reg #2.
                        ;*              TXBFCA - Contains address of DMA character count register.
                        ;*
                        ;* OUTPUTS:     CARRY - Success flag (set if DMA_START found clear).
                        ;*              DUT TBUFFAD1 - LS 16 bits of DMA buffer address (initialized).
                        ;*              DUT TBUFFAD2 - MS 6 bits of DMA buffer address (initialized),
                        ;*                             DMA_START bit set.
                        ;*              DUT TBUFFCT - DMA buffer character count (initialized).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,DODMA
                        ;*
                        ;* COMMENTS:    This routine determines if Memory Management is being used
                        ;*              and sets up the full 22 bit physical address if necessary.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ************************************************************************

013760                  DODMA:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
013760  004567  170130                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
013764  012704  000200          MOV     #200,R4         ;PREPARE TO CLEAR UPPER 6 BITS OF DMA BUFF ADR.
013770  005767  166352          TST     MMENAB          ;CHECK FOR MEMORY MANAGEMENT IN USE.
013774  001427                BEQ     6$              ;GOTO SET UP DEVICE IF MEM MGT NOT IN USE.
                        ;+
                        ; Memmory management is in use.
                        ; Construct 22 bit physical address from the 16 bit virtual address.
                        ;-
013776  010205          MOV     R2,R5           ;STRIP THE MOST SIGNIFICANT 3 BITS OF THE
014000  012700  000005          MOV     #5,R0           ;  DMA BUFFER VIRTUAL ADDRESS AND MULTIPLY
014004  006105          2$:     ROL     R5              ;  THEIR VALUE BY TWO TO GET AN OFFSET INTO
014006  005300                  DEC     R0              ;  THE TABLE OF MEMMORY MANAGEMENT PAGE
014010  001375                BNE     2$              ;  ADDRESS REGISTERS (PAR).
014012  042705  177761          BIC     #177761,R5      ;
014016  066705  166326          ADD     PAR0A,R5        ;ADD IN THE BASE VALUE OF THE MM PAR REGISTERS.
014022  011505                  MOV     (R5),R5         ;GET THE 16 BIT PHYSICAL ADDRESS BLOCK COUNT.
014024  012700  000006          MOV     #6,R0           ;SHIFT UPPER 6 BITS OF THE PHYSICAL ADDRESS
014030  006305          4$:     ASL     R5              ;  BLOCK COUNT (GOTTEN FROM THE PROPER PAR)
014032  006104                  ROL     R4              ;  INTO THE LS 6 BITS OF THE WORD TO WRITE
014034  005300                  DEC     R0              ;  INTO THE DUT TBUFFAD2 REGISTER.
014036  001374                BNE     4$              ;
014040  042702  160000          BIC     #160000,R2      ;ADD THE 13 BIT DISPLACEMENT FIELD FROM VIRTUAL
014044  060502                  ADD     R5,R2           ;  ADR TO THE SHIFTED BLOCK NUMBER FROM THE
014046  005504                  ADC     R4              ;  MEMMORY MANAGEMENT PAR.
014050  052704  000200          BIS     #200,R4         ;SET THE DMA_START BIT IN WORD FOR TBUFFAD2.
```

GLOBAL SUBROUTINE - DODMA -

```
                                ;+
                                ; Write the DMA parameters out to the DUT DMA registers.
                                ;    Disable interrupts.
                                ;    Set up DUT CSR IND.ADR.REG field.
                                ;    Write the DMA transmit character count.
                                ;    Write the least significant 16 bits of the DMA buffer start address.
                                ;    Write the most significant 6 bits of the address,
                                ;      setting the DMA_START bit, and initiating the DMA transmission.
                                ;-
014054                          6$:     GETPRI  R5              ;GET THE PRESENT PROCESSOR PRIORITY.
014054  104440                                                          TRAP    C$GPRI
014056  010005                                                          MOV     R0,R5
014060                                  SETPRI  #PRI07          ;DISABLE ALL HARDWARE INTERRUPTS.
014060  012700  000340                                                  MOV     #PRI07,R0
014064  104441                                                          TRAP    C$SPRI
014066  056701  166176                  BIS     IESTAT,R1       ;PREPARE FOR SETUP OF LINE NUMBER IN DUT CSR.
014072  010177  166144                  MOV     R1,@CSRA        ;SET UP THE DUT CSR IND.ADR.REG FIELD.
014076  105777  166154                  TSTB    @TXAD2A         ;TEST THE DUT DMA_START BIT.
014102  000241                          CLC                     ;INDICATE FAILURE IN CASE DMA.HO BIT IS SET.
014104  100411                         BMI     60$             ;EXIT WITH FAILURE IF DMA.HO BIT IS SET.
014106  010377  166146                  MOV     R3,@TXBFCA      ;WRITE THE DMA CHARACTER COUNT.
014112  010277  166136                  MOV     R2,@TXAD1A      ;WRITE THE LS 16 BITS OF BUFFER ADDRESS.
014116  110477  166134                  MOVB    R4,@TXAD2A      ;WRITE MS 6 BITS OF ADR AND START DMA TX.
014122                                 SETPRI  R5              ;RESTORE THE PROCESSOR PRIORITY.
014122  010500                                                          MOV     R5,R0
014124  104441                                                          TRAP    C$SPRI
014126  000261                          SEC                     ;INDICATE SUCCESS.

014130                          60$:    PASS                    ;RESTORE GPRS,
014130  004736                                  JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
014132  000207                          RTS     PC              ;  CARRY - SUCCESS FLAG (SET IF SUCCESS).
```

GLOBAL SUBROUTINE - FINACT -

```
                                .SBTTL  GLOBAL SUBROUTINE                - FINACT -
                                ;++ ************************************************************************
                                ;*                        - FIND FIRST ACTIVE LINE -
                                ;*      This subroutine calculates the number of the first active line that
                                ;*      is found in the active line bit map ACTLNS.
                                ;*
                                ;* INPUTS:      ACTLNS - Contains the active line bit map.
                                ;*
                                ;* OUTPUTS:     R1 - Contains the number of the first active line.
                                ;*              R5 - Contains the bit map representation of the active line.
                                ;*              Carry set indicates success.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,FINACT
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************
014134                          FINACT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014134  004567  167754                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                    ;+
                                    ; Find an active line on which to perform the test.
                                    ;-
014140  005001                          CLR     R1                      ;CLEAR THE LINE NUMBER COUNTER.
014142  012703  000010                  MOV     #NUMLNS,R3              ;GET MAX LINE NUMBER.
014146  016700  166062                  MOV     ACTLNS,R0               ;GET THE ACTIVE LINE BIT MAP.
014152  012705  000001                  MOV     #1,R5                   ;SET UP A LINE BIT MASK.
014156  030500                  2$:     BIT     R5,R0                   ;LOOK FOR AN ACTIVE LINE.
                                BNE     4$                              ;BRANCH TO BEGIN TEST IF A LINE HAS BEEN FOUND.
                                 ASL    R5                              ;SHIFT THE BIT MASK FOR THE NEXT LINE.
                                 INC    R1                              ;INCREMENT THE LINE NUMBER COUNTER.
                                 CMP    R1,R3                           ;CHECK IF ALL LINES HAVE BEEN TRIED.
                                BLT     2$                              ;LOOP TO TRY THE NEXT LINE.
                                 CLC                                    ;CLEAR CARRY BIT, NO ACTIVE LINE FOUND.
                                BR      60$                             ;EXIT WITH FAILURE.
                                4$:      SEC                            ;SET CARRY, SUCCESS.

014200                          60$:    PASS    R1,R5                   ;RESTORE GPRS, EXCEPT
014200  010166  C00004                                  MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
014204  010566  000014                                  MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
014210  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                ;R1 - CONTAINS THE NUMBER OF FIRST ACTIVE LINE.
                                ;R5 - CONTAINS THE BIT MAP OF THE ACTIVE LINE.
                                ;CARRY - SET INDICATES SUCCESS.
014212  000207                  RTS     PC
```

GLOBAL SUBROUTINE - INDATP -

```
                        .SBTTL  GLOBAL SUBROUTINE                - INDATP -
                        ;++ ************************************************************************
                        ;*                          - INITIALISE DATA PATTERN -
                        ;*      This subroutine is used to initialise an incremental byte data pattern
                        ;*      in the general buffer area.
                        ;*      The data pattern will be sequential from 0 to 255 (decimal).
                        ;*
                        ;* INPUTS:      BUFBAS - Address of the start of the general buffer area.
                        ;*              BUFMID - Address of the 255 th location.
                        ;*
                        ;* OUTPUTS:     The first 255 locations of the general buffer area contain data
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,INIDATP
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

014214                  INDATP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014214  004567  167674                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

014220  012702  002704          MOV     #BUFBAS,R2      ;INITIALIZE THE DATA PATTERN IN THE GENERAL
014224  005003                  CLR     R3              ;  DATA BUFFER TO A 256 BYTE PATTERN.
014226  110322          2$:     MOVB    R3,(R2)+        ;
014230  005203                  INC     R3              ;SELECT THE NEXT CHARACTER.
014232  020227  003304          CMP     R2,#BUFMID      ;CHECK IF WE HAVE 256 DATA PATTERNS.
014236  103773                  BLO     2$              ;

014240                  60$:    PASS                    ;RESTORE GPRS.
014240  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014242  000207                  RTS     PC
```

GLOBAL SUBROUTINE - INDTPX -

```
                                .SBTTL  GLOBAL SUBROUTINE               - INDTPX -
                                ;++ ************************************************************************
                                ;*              - INITIALISE DATA PATTERN WITHOUT XON OR XOFF -
                                ;*      This subroutine is used to initialise an incremental byte data pattern
                                ;*      in the general buffer area.
                                ;*      The data pattern will be from 0 to 255, but will exclude the following
                                ;*      two characters; (ASCII DC1, DC3) XON AND XOFF.  This will cause the
                                ;*      last two data characters to be the same as the first two.
                                ;*
                                ;* INPUTS:      BUFBAS - Address of the start of the general buffer area.
                                ;*              BUFMID - Address of the 255 th location.
                                ;*
                                ;* OUTPUTS:     The first 255 locations of the general buffer area contain data
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,INDTPX
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

014244                          INDTPX:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014244  004567  167644                                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;+
                                ; Initialize the 256 byte data pattern.
                                ;  Ensure the data pattern is free from XON's or XOFF's to prevent errors.
                                ;  Note: the first two characters and the last two characters will be the same.
                                ;-
014250  012702  002704                  MOV     #BUFBAS,R2      ;INITIALIZE THE DATA PATTERN IN THE GENERAL
014254  005003                          CLR     R3              ;  DATA BUFFER TO A 256 BYTE PATTERN.
014256  110322                  2$:     MOVB    R3,(R2)+        ;
014260  105203                          INCB    R3              ;SELECT THE NEXT CHARACTER.
014262  122703  000021                  CMPB    #21,R3          ;CHECK FOR AN XON CHARACTER.
014266  001001                          BNE     4$              ;BRANCH IF CHAR NOT AN XON.
014270  105203                          INCB    R3              ;FORCE THE NEXT CHARACTER.
014272  122703  000023          4$:     CMPB    #23,R3          ;CHECK FOR AN XOFF CHARACTER.
014276  001001                          BNE     6$              ;BRANCH IF NOT AN XOFF CHARACTER.
014300  105203                          INCB    R3              ;FORCE THE NEXT CHARACTER.
014302  020227  003304          6$:     CMP     R2,#BUFMID      ;CHECK IF WE HAVE 256 DATA PATTERNS.
014306  103763                          BLO     2$              ;

014310                          60$:    PASS                            ;RESTORE GPRS.
014310  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014312  000207                          RTS     PC
```

GLOBAL SUBROUTINE - LINBIT -

```
                                .SBTTL  GLOBAL SUBROUTINE                - LINBIT -
                                ;++ ************************************************************************
                                ;*              - Line Number to Bit Map conversion subroutine -
                                ;*      This subroutine is used to generate a bit map (one bit of 16 set)
                                ;*      based on a line number (range: 1 to 16).  Only the LS 4 bits of the
                                ;*      line number word are used, the others are masked out (so unmasked
                                ;*      MSBytes of DUT CSRs can be passed to this routine without error).
                                ;*
                                ;* INPUTS:      R1 - Line number (only LS 4 bits used, others disregarded).
                                ;*              BITTBL - Base label of a 16 word bit table.
                                ;*
                                ;* OUTPUTS:     R0 - Bit map, bit corresponding to line number is set:
                                ;*                      If line number is 3, then bit3 is set, etc.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,LINBIT
                                ;*
                                ;* COMMENTS:    No checking is performed to verify that the line number is
                                ;*              a legal line number for the DUT (ie - less than NUMLNS).
                                ;*              NOTE: The line number is not destroyed of altered, so this
                                ;*                      routine can be used easily in loops.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;-- ************************************************************************

014314                          LINBIT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014314  004567  167574                           JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014320  042701  177760                 BIC     #177760,R1              ;MASK OUT ALL BUT 4 LSBITS OF THE LINE #.
014324  006301                         ASL     R1                      ;MULTIPLY LINE # BY 2 TO GET WORD TABLE OFFSET.
014326  016100  002370                 MOV     BITTBL(R1),R0           ;GET THE SINGLE BIT BIT MAP.
014332                          60$:    PASS    R0                      ;RESTORE GPRS, EXCEPT THE FOLLOWING,
014332  010066  000002                                 MOV     R0,ROSLOT(SP)           ;PUT R0 IN STACK SLOT.
014336  004736                                         JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014340  000207                         RTS     PC                        ;R0 - BIT MAP WITH LINE # BIT SET.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - MAPCNT -
                        ;** ************************************************************************
                        ;*                          - Count Bits in Bit Map Routine -
                        ;*      This subroutine counts the number of bits which are set in a bit map.
                        ;*
                        ;* INPUTS:      R2 - The bit map for which to count the bits.
                        ;*
                        ;* OUTPUTS:     R2 - Count of the number of bits that were set.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,MAPCNT
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ************************************************************************

014342                  MAPCNT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014342  004567  167546                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014346  010201                  MOV     R2,R1
014350  001405                BEQ     60$             ;EXIT WITH ZERO IF NO BITS ARE SET IN MAP.

014352  005002                  CLR     R2              ;CLEAR THE BIT COUNT.
014354  000261                  SEC                     ;COUNT THE LAST BIT TO BE SHIFTED OUT.

014356  005502          2$:     ADC     R2              ;COUNT THE BIT IF IT WAS SET.
014360  006301                  ASL     R1              ;SHIFT ANOTHER BIT OUT OF THE MAP.
014362  001375                BNE     2$              ;LOOP IF ALL BITS NOT SHIFTED OUT OF MAP.

014364                  60$:    PASS    R2              ;RESTORE GPRS, EXCEPT THE FOLLOWING:
014364  010266  000006                         MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
014370  004736                               JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014372  000207                RTS     PC              ;  R2 - COUNT OF BITS SET IN BIT MAP.
```

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

014374                  MSLGET:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014374  004567  167514                          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.
                        ;-
014400  005102                  COM     R2              ;GET MASK OF UNUSED BITS.
014402  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.
                        ;-
014404  005701                  TST     R1              ;TEST THE TIME-OUT VALUE FOR ZERO.
014406  001011                  BNE     2$              ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
014410  011400                  MOV     (R4),R0         ;GET THE WORD TO TEST BEFORE EXITING.
014412  010067  000070          MOV     R0,62$          ;SAVE VALUE SO WE CAN RETURN IT.
014416  040200                  BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
014420  020003                  CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
014422  000261                  SEC                     ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
014424  001420                  BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
```

GLOBAL SUBROUTINE - MSLGET -

```
3132 014426  000241                    CLC                     ;INDICATE FAILURE (TIME-OUT).
3133 014430  000416                    BR      6$              ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
3134                                ;+
3135                                ; Non-zero time-out value.  Loop, waiting for condition or time-out.
3136                                ;-
3137 014432  016705  165702        2$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
3138 014436  011400                4$:     MOV     (R4),R0         ;GET THE WORD TO TEST.
3139 014440  010067  000042                MOV     R0,62$          ;SAVE WORD IN CASE THIS IS THE LAST.
3140 014444  040200                        BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
3141 014446  020003                        CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
3142 014450  000261                        SEC                     ;SET CARRY IN CASE OF SUCCESS.
3143 014452  001405                        BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
3144 014454  005305                        DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
3145 014456  001367                        BNE     4$              ;LOOP IF MS NOT UP.
3146 014460  005301                        DEC     R1              ;DECREMENT THE MS TIME COUNT.
3147 014462  001363                        BNE     2$              ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
3148 014464  000241                        CLC                     ;CLEAR CARRY, WE TIMED-OUT.
3149                                ;+
3150                                ; Have either found condition, or timed-out (possibly from 0 time-out value).
3151                                ; Restore the last contents read from the test word.  Exit routine.
3152                                ;-
3153 014466  016700  000014        6$:     MOV     62$,R0          ;PASS OUT THE LAST READ WORD.
3154 014472                        60$:    PASS    R0,R1           ;RESTORE GPRS, EXCEPT THE FOLLOWING:
     014472  010066  000002                        MOV     R0,R0SLOT(SP)       ;PUT R0 IN STACK SLOT.
     014476  010166  000004                        MOV     R1,R1SLOT(SP)       ;PUT R1 IN STACK SLOT.
     014502  004736                                JSR     PC,@(SP)+           ;RETURN TO PREGO5 SUBRT.
3155                                                    ;R0 - LAST READ WORD CHECKED FOR CONDITION.
3156                                                    ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
3157 014504  000207                        RTS     PC              ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
3158                                ;+
3159                                ; Local storage.
3160                                ;-
3161 014506  000000                62$:    .WORD   0               ;STORAGE FOR THE LAST READ WORD.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - MSLOOP -
                        ;*****************************************************************************
                        ;*      - Test Loop subroutine -
                        ;*      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 thereafter.
                        ;*
                        ;* 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:     CARRY - Success flag (set if condition is met before time-out).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,MSLOOP
                        ;*
                        ;* 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: MSLGET.
                        ;*****************************************************************************

014510                  MSLOOP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014510  004567  167400                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

                        ;+
                        ; Calling the MSLGET routine from the MSLOOP routine isolates the caller of
                        ; MSLOOP from the returned test word and remaining time-out values.
                        ;-
014514  004767  177654          JSR     PC,MSLGET               ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.

014520                  60$:    PASS                            ;RESTORE GPRS,
014520  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014522  000207                  RTS     PC                        ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
```

GLOBAL SUBROUTINE - OOPS -

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

014524                          OOPS::  SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
014524  004567  167364                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                            ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
014530                          ERRSF   101,EM0101
014530  104454                                                          TRAP    C$ERSF
014532  000145                                                          .WORD   101
014534  014570                                                          .WORD   EM0101
014536  000000                                                          .WORD   0
                            ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
014540                          PRINTF  #EM0102
014540  012746  014654                                                  MOV     #EM0102,-(SP)
014544  012746  000001                                                  MOV     #1,-(SP)
014550  010600                                                          MOV     SP,R0
014552  104417                                                          TRAP    C$PNTF
014554  062706  000004                                                  ADD     #4,SP
014560                  2$:     BREAK                   ;LOOK FOR OPERATOR CONTROL-C INPUT.
014560  104422                                                          TRAP    C$BRK
014562  000776                  BR      2$              ;INFINITE LOOP.
014564                  60$:    PASS                    ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
014564  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014566  000207                  RTS     PC              ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.

014570     110     117     123  EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
014573     124     040     103
014576     117     115     120
014601     125     124     105
014604     122     040     110
014607     101     122     104
014612     127     101     122
014615     105     040     117
014620     122     040     123
014623     117     106     124
014626     127     101     122
014631     105     040     102
014634     125     107     040
014637     105     116     103
014642     117     125     116
014645     124     105     122
```

```
GLOBAL SUBROUTINE                - OOPS -

        014650      105      104      056
        014653      000
   3237 014654      045      116      045  EM0102:: .ASCIZ /%N%APROGRAM HUNG, WAITING FOR A CONTROL-C. <**************%N%N/
        014657      101      120      122
        014662      117      107      122
        014665      101      115      040
        014670      110      125      116
        014673      107      054      040
        014676      127      101      111
        014701      124      111      116
        014704      107      040      106
        014707      117      122      040
        014712      101      040      103
        014715      117      116      124
        014720      122      117      114
        014723      055      103      056
        014726      040      074      052
        014731      052      052      052
        014734      052      052      052
        014737      052      052      052
        014742      052      052      052
        014745      045      116      045
        014750      116      000
   3238                                      .EVEN
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - PRTLPR -
                                ;++ ************************************************************************
                                ;*                        -Print the contents of the LPR.
                                ;*      This routine is used to print out extended information on the
                                ;*      contents of the Line Parameter Register (LPR).
                                ;*
                                ;* INPUTS:      R3 - Contains the number of the line you wish to examine.
                                ;*              CSRA - Contains the address of the DUT's CSR.
                                ;*              IESTAT - Contains the current status of the TX and RX interrupt
                                ;*                       enable bits in the DUT's CSR.
                                ;*              LPRA - Contains the address of the DUT's LPR register.
                                ;*
                                ;* OUTPUTS:     An extended information message is printed on the operators
                                ;*              console.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,PRTLPR
                                ;*
                                ;* COMMENTS:    This routine changes the indirect address field of the device
                                ;*              under test's CSR.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

014752                          PRTLPR::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
014752  004567  167136                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014756  016701  165260                 MOV     CSRA,R1         ;GET THE CSR ADDRESS.
014762  016702  165260                 MOV     LPRA,R2         ;GET THE LPR ADDRESS.
014766  042703  177760                 BIC     #177760,R3      ;CLEAR ANY UNWANTED BITS.
014772  056703  165272                 BIS     IESTAT,R3       ;SET STATE OF TX AND RX INTERRUPT ENABLE BITS.
014776  010311                       MOV     R3,(R1)         ;SELECT LINE.
015000  011204                       MOV     (R2),R4         ;GET CONTENTS OF THE LPR.
                                    ;PRINT MESSAGE"CONTENTS OF THE LPR:nnnnnn"
015002                              PRINTX #EF9019,#EM9026,R4;PRINT OUT MESSAGE ON OPERATORS CONSOLE.
015002  010446                                                                  MOV     R4,-(SP)
015004  012746  011027                                                          MOV     #EM9026,-(SP)
015010  012746  005276                                                          MOV     #EF9019,-(SP)
015014  012746  000003                                                          MOV     #3,-(SP)
015020  010600                                                                  MOV     SP,R0
015022  104415                                                                  TRAP    C$PNTX
015024  062706  000010                                                          ADD     #10,SP
015030                          60$:    PASS                    ;RESTORE GPRS.
015030  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
015032  000207                          RTS     PC
```

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

015034                          PUFIFO::SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
015034  004567  167054                                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015040  012701  001000                  MOV     #512.,R1                ;SET MAXIMUM TRY COUNT OF 512.
015044  016704  165174                  MOV     RBUFA,R4                ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.

015050  011402                  2$:     MOV     (R4),R2                 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
015052  100016                         BPL     6$                      ;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.
                                ;-
015054  012700  070000                  MOV     #70000,R0               ;GENERATE A BIT MAP OF CHAR ERROR BITS
015060  040200                          BIC     R2,R0                   ; WHICH ARE NOT SET FOR CHAR.
015062  001006                         BNE     4$                      ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
                                  ;+
                                  ; Check if the read data is modem status , BMP or Selftest?.
                                  ;-
015064  012700  000300                  MOV     #300,R0                 ; CHECK IF BMP OR SELFTEST?.
015070  040200                          BIC     R2,R0                   ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
015072  001002                         BNE     4$                      ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
015074  004767  000444                 JSR     PC,SAVBMP               ;SAVE BMP CODE ON THE QUEUE.

015100  005301                  4$:     DEC     R1                      ;DECREMENT THE TRY COUNT.
015102  001362                         BNE     2$                      ;LOOP TO TRY AGAIN.
015104  000241                          CLC                             ;CLEAR CARRY,TO INDICATE FIFO NOT PURGED.
015106  000401                         BR      60$                     ;EXIT WITH CARRY CLEAR.
015110  000261                  6$:     SEC                             ;SET CARRY, TO INDICATE FIFO PURGED.

015112                          60$:    PASS                            ;RESTORE GPRS,
015112  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                        ;CARRY BIT, SET INDICATES FIFO PURGED.
015114  000207                         RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - PUFIFR -
                        ;*************************************************************************
                        ;*      - Purge FIFO report any errors found.
                        ;*      This routine removes all data from the FIFO.  Any BMP codes that are
                        ;*      found are save on the queue to be reported later in the BMP report test.
                        ;*      Any unxepected data (ie any non-status inforamtion) that are found,
                        ;*      are reported as an error.
                        ;*      If the FIFO will not purge after 512 attempts, then the current test
                        ;*      that called this routine receives a failure flag that should be used
                        ;*      to abort the test.
                        ;*
                        ;* INPUTS:      ERRTBL - ERRTYPE, ERRMSG, ERRNBR are set up correctly.
                        ;*              RBUFA- Contains the  address of the Receiver.
                        ;*
                        ;* OUTPUTS:     Carry bit - Abort test flag, Clr = ABORT TEST, Set = OK.
                        ;*              ERRBLK - Value will be dastroyed.
                        ;*              BMPCQP - The BMP code queue pointer may be updated.
                        ;*              The contents of the BMP code queue may be udated.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,PUFIFR
                        ;*
                        ;* COMMENTS:    This routine reports errors with numbers initial ERRNBR
                        ;*              thru to ERRNBR+2.
                        ;*              The ERRNBR is restored to its INITIAL value before returning.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ER1603,ER9001,ER9002,SAVBMP.
                        ;*************************************************************************

015116                  PUFIFR::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
015116  004567  166772                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015122  016746  166760          MOV     ERRNBR,-(SP)    ;SAVE THE CONTENTS OF THE ERROR NUMBER.
015126  012705  001000          MOV     #512.,R5        ;SET MAXIMUM READ COUNTER TO 2*FIFO SIZE.
                        ;+
                        ; Read data from the FIFO until DATA VALID is clear of read counter is zero.
                        ;  Report any BMP or Unexpected data as errors.
                        ;-
015132  017702  165106  2$:     MOV     @RBUFA,R2       ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
015136  100057          BPL     8$              ;EXIT IF DATA VALID CLEAR, ie. FIFO PURGED.
                         ;+
                         ; Check if read data is status or unexpected character.
                         ;-
015140  012700  070000          MOV     #70000,R0       ;GENERATE A BIT MAP OF CHAR ERROR BITS
015144  040200          BIC     R2,R0           ; WHICH ARE NOT SET FOR CHAR.
015146  001012          BNE     4$              ;SKIP BMP CHECK IF IT IS UNEXPECTED DATA.
                         ;+
                         ; Check if the read data is modem status , BMP or Selftest?.
                         ;  If it is a BMP code then save it on the queue.
                         ;-
015150  012767  012334  166734          MOV     #ER9001,ERRBLK  ;SET UP THE CORRECT ERROR REPORTING ROUTINE.
015156  012700  000300          MOV     #300,R0         ; CHECK IF BMP OR SELFTEST?.
015162  040200          BIC     R2,R0           ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
015164  001003          BNE     4$              ;SKIP BMP ERROR REPORT IF MODEM OR SELFTEST?.
015166  004767  000352          JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
015172  000424          BR      6$              ;BRANCH TO CHECK READ COUNT.
                         ;+
                         ; Check if the read data is Modem, Selftest or Unexpected data.
                         ;-
```

GLOBAL SUBROUTINE - PUFIFR -

```
3381 015174  032702  000001          4$:     BIT     #BIT0,R2        ;TEST THE MODEM STATUS INDICATION BIT.
3382 015200  001421                          BEQ     6$              ;DO NOT REPORT ANY ERROR IF MODEM STATUS.
3383 015202  012701  011053                  MOV     #EM9104,R1      ;PASS THE CORRECT ERROR MESSAGE TO REPORT.
3384 015206  010203                          MOV     R2,R3           ;EXTRACT THE LINE NUMBER FROM
3385 015210  000303                          SWAB    R3              ; THE READ DATA.
3386 015212  042703  177760                  BIC     #177760,R3      ;
3387 015216  006303                          ASL     R3              ;FORM LINE NUMBER TIMES 2 FOR ER9002 ROUTINE.
3388 015220  052704  100000                  BIS     #BIT15,R4       ;SET THE "NONE" EXPECTED MAESSAGE FLAG.
3389 015224  005267  166656                  INC     ERRNBR          ;SET ERROR NUMBER TO INTIAL ERRBR+1.
3390 015230  012767  012424  166654          MOV     #ER9002,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
3391                                         ;REPORT ERROR "UNEXPECTED DATA FOUND IN FIFO".
3392 015236                                  ERROR                   ;               >>>>> ERROR <<<<<.
     015236  104460                                                                  TRAP    C$ERROR
3393 015240  005367  166642                  DEC     ERRNBR          ;RESTORE ERROR NUMBER TO INTIAL ERRNBR.
3394
3395 015244  005305                  6$:     DEC     R5              ;DECREMENT READ COUNTER.
3396 015246  001331                          BNE     2$              ;LOOP TO READ NEXT CHAR FROM FIFO IF COUNT > 0.
3397                                  ;+
3398                                  ; The FIFO will not clear, report the error and indicate that the test is to
3399                                  ; be ABORTED.
3400                                  ;-
3401 015250  062767  000002  166630          ADD     #2,ERRNBR       ;SET ERROR NUMBER TO INTIAL ERRNBR+2.
3402 015256  012767  011662  166626          MOV     #ER1603,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
3403 015264  012701  010716                  MOV     #EM9017,R1      ;PASS THE MESSAGE TO BE REPORTED.
3404                                         ;REPORT THE ERROR "FIFO WILL NOT PURGE, (DATA VALID STUCK SET)"
3405                                         ;                       "?????? TEST ABORTED".
3406 015270                                  ERROR                   ;               >>>>> ERROR <<<<<.
     015270  104460                                                                  TRAP    C$ERROR
3407 015272  000241                          CLC                     ;INDICATE THE TEST IS TO BE ABORTED.
3408 015274  000401                          BR      10$             ;EXIT THIS ROUTINE AND ABORT THE CURRENT TEST.
3409
3410 015276  000261                  8$:     SEC                     ;SET THE CARRY, DO NOT ABORT THE TEST.
3411
3412 015300  012667  166602          10$:    MOV     (SP)+,ERRNBR    ;RESTORE INITIAL ERROR NUMBER.
3413 015304                          60$:    PASS                    ;RESTORE GPRS,
     015304  004736                                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3414                                         ;CARRY BIT, SET INDICATES FIFO PURGED, DO NOT
3415                                         ; ABORT THE TEST.
3416 015306  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - READBX -
                        ;** ************************************************************************
                        ;*       - READ CHARACTERS FROM THE FIFO AND CHECKS FOR BMPS AND XONS-
                        ;*       This subroutine is used in the FIHAVL.TST.
                        ;*       It reads the specified number of characters from the fifo and checks
                        ;*       for BMP codes and XON characters.
                        ;*
                        ;* INPUTS:       R0 - Contains the number of chars to read from the fifo.
                        ;*
                        ;* OUTPUTS:      R1 - Contains address of error message to be reported
                        ;*                    Clear if no error found.
                        ;*               Carry used to indicate if fifo was found empty, carry clear.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,READ
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: CHKBMP.
                        ;-- ************************************************************************

015310                  READBX:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
015310  004567  166600                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015314  005001                  CLR     R1                      ;CLEAR GPR THAT HOLDS THE ADDRESS OF ERRMSG.
015316  016703  164722          MOV     RBUFA,R3                ;GET THE ADDRESS OF THE RECEIVER BUFFER REG.
015322  011302                  2$:     MOV     (R3),R2         ;READ A CHARACTER FROM THE FIFO.
015324  100015                  BPL     8$                      ;BRANCH IF FIFO IS EMPTY.
                          ;*
                          ; Check if the read character is a BMP code.
                          ;  If it is a BMP code save it on the queue to be reported later, and
                          ;  abort the test.
                          ;-
015326  004767  176054              JSR     PC,CHKBMP           ;CHECK IF CHARACTER IS A BMP CODE.
015332  103410                    BCS     6$                  ;BRANCH IF A BMP CODE WAS FOUND.
015334  120227  000021             CMPB    R2,#21              ;CHECK IF IT IS AN XON.
015340  001003                    BNE     4$                  ;BRANCH IF NOT AN XON.
015342  012701  006744             MOV     #EM5402,R1          ;PASS THE MESSAGE TO BE REPORTED.
015346  000402                    BR      6$                  ;GO EXIT TEST.
015350  005300              4$:     DEC     R0                  ;DECREMENT THE READ COUNT.
015352  001363                    BNE     2$
015354  000261              6$:     SEC                         ;SET CARRY TO INDICATE SUCCESS.
015356  000401                    BR      60$                 ;EXIT
015360  000241              8$:     CLC                         ;CLEAR CARRY BIT TO INDICATE FAILURE.

015362                      60$:    PASS    R1                  ;RESTORE GPRS,
015362  010166  000004                                  MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
015366  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
015370  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.
                        ;*************************************************************************

015372                  RESETT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
015372  004567  166516                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015376  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.
                        ;-
015402  016704  164634          MOV     CSRA,R4         ;GET THE ADDRESS OF THE DUT'S CSR.
015406  030214                  BIT     R2,(R4)         ;CHECK STATE OF MASTER RESET BIT.
015410  001406                 BEQ     2$              ;DON'T DELAY IF MR IS ALREADY CLEAR.
015412  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
015414  012701  004704          MOV     #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
015420  004767  176750         JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
015424  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.
                        ;-
015426  010277  164610  2$:      MOV     R2,@CSRA        ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
015432  004767  000266         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.
                          ;-
015436  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
015440  012701  004704          MOV     #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
015444  004767  176724         JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
015450  103410                 BCS     6$              ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
```

GLOBAL SUBROUTINE - RESETT -

```
3519                                    ;+
3520                                    ; Set up error message to report "fatal error found during reset,test aborted".
3521                                    ; Indicate test is to be aborted by clearing the carry bit.
3522                                    ;-
3523 015452  012701  005711            4$:     MOV     #EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
3524 015456  012767  011662  166426            MOV     #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
3525                                            ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
3526                                            ; "TEST ABORTED"
3527 015464                                    ERROR                   ;               >>>>> ERROR <<<<<
     015464  104460                                                                            TRAP    C$ERROR
3528 015466  000241                            CLC                     ;INDICATE TEST IS TO BE ABORTED.
3529 015470  000403                            BR      60$             ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
3530                                    ;+
3531                                    ; Clear TX and RX Interrupt enable status flags in IESTAT.
3532                                    ; Exit with continue test indicator set (ie,carry set).
3533                                    ;-
3534 015472  005067  164572            6$:      CLR    IESTAT          ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
3535 015476  000261                             SEC                    ;INDICATE SUCCESS, CONTINUE TEST.
3536
3537 015500                            60$:    PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
     015500  004736                                            JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3538                                            ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
3539 015502  000207                            RTS     PC
3540
```

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.
                                ;-- ************************************************************************
015504  010046                RXIE0:: MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
015506                                  GETPRI -(SP)            ;SAVE PROCESSOR PRIORITY ON STACK.
015506  104440                                                                    TRAP    C$GPRI
015510  010046                                                                    MOV     R0,-(SP)
015512                                  SETPRI #PRI07           ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
015512  012700  000340                                                            MOV     #PRI07,R0
015516  104441                                                                    TRAP    C$SPRI
015520  042767  137777  164542        BIC     #137777,IESTAT  ;CLEAR RX.INT.ENBL BIT IN IESTAT.
015526  016777  164536  164506        MOV     IESTAT,@CSRA    ;DISABLE RX INTERRUPTS.
015534                                  SETPRI (SP)+            ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
015534  012600                                                                    MOV     (SP)+,R0
015536  104441                                                                    TRAP    C$SPRI
015540  012600                          MOV     (SP)+,R0        ;RESTORE R0.
015542  000207                        RTS     PC
```

GLOBAL SUBROUTINE - SAVBMP -

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

015544                          SAVBMP:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
015544  004567  166344                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015550  016704  164666                 MOV     BMPCQP,R4       ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
015554  116724  164506                 MOVB    TSTNUM,(R4)+    ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
015560  005204                         INC     R4              ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
015562  042702  177400                 BIC     #177400,R2      ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
015566  010224                         MOV     R2,(R4)+        ;SAVE THE BMP CODE ON THE QUEUE.
015570  020427  002644                 CMP     R4,#BMPCQE      ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
015574  103402                       BLO     2$              ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
015576  162704  000004                 SUB     #4,R4           ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
015602  010467  164634         2$:     MOV     R4,BMPCQP       ;SAVE THE POINTER.

015606                          60$:    PASS                    ;RESTORE GPRS.
015606  004736                                         JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
015610  000207                       RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - SAVMST -
                                ;++ ************************************************************************
                                ;*                        - Save Modem Status Routine -
                                ;*      This routine saves the present contents of the DUT STAT registers in
                                ;*      the STAT storage table.
                                ;*
                                ;* INPUTS:      CSRA - Contains the address of the DUT CSR.
                                ;*              IESTAT - State of the DUT CSR interrupt enable bits.
                                ;*              NUMLNS - Equated to the number of lines on the DUT.
                                ;*              STATA - Contains the address of the DUT STAT register.
                                ;*              STSTB - Label at base of the STAT storage table.
                                ;*
                                ;* OUTPUTS:     STST Table - Overwritten with present STAT contents.
                                ;*              CSR IND.ADR.REG field - Destroyed.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,SAVMST
                                ;*
                                ;* COMMENTS:    If the contents of IESTAT changes during this test the CSR
                                ;*              interrupt enable bits will not track the change.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;-- ************************************************************************

015612                          SAVMST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
015612  004567  166276                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015616  016701  164446                 MOV     IESTAT,R1               ;GET IE STATES FOR UPDATING IND.ADR.REG FIELD.
015622  012702  002644                 MOV     #STSTB,R2               ;SET UP STAT STORAGE POINTER TO BASE OF TABLE.
015626  012703  000010                 MOV     #NUMLNS,R3
015632  050103                         BIS     R1,R3                   ;FORM COMPLETION COMPARISON WORD.
015634  010177  164402         2$:     MOV     R1,@CSRA                ;SET UP THE CSR IND.ADR.REG FIELD.
015640  017722  164404                 MOV     @STATA,(R2)+            ;SAVE CONTENTS OF THIS LINE'S STAT REGISTER.
015644  005201                         INC     R1                      ;SET LINE COUNTER TO NEXT LINE.
015646  020103                         CMP     R1,R3                   ;CHECK FOR ALL LINES DONE.
015650  002771                       BLT     2$                        ;LOOP IF NOT ALL LINES DONE.

015652                          60$:    PASS                            ;RESTORE GPRS.
015652  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
015654  000207                       RTS     PC
```

GLOBAL SUBROUTINE - SETPAR -

```
                                .SBTTL  GLOBAL SUBROUTINE                - SETPAR -
                                ;++ ************************************************************************
                                ;*                        - SET TX AND CONTROL PARAMETERS -
                                ;*      This suroutine is used in the FIHAVL.TST.
                                ;*      It initialises the selected line to the following state:
                                ;*      Internal loopback, IAUTO enabled, LPR:38.4k, 8 bits/char, 2 stop,
                                ;*      odd parity.
                                ;*
                                ;* INPUTS:      R1 - Contains number of the line to be initialised.
                                ;*
                                ;* OUTPUTS:     LNCTRL and LPR registers for the selected line are destroyed.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,SETPAR
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: DELAY,WTWLNC,WTWLPR.
                                ;-- ************************************************************************

015656                          SETPAR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
015656  004567  166232                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015662  004767  176426                 JSR     PC,LINBIT       ;GET A BIT MAP FOR THIS LINE.
015666  010005                          MOV     R0,R5           ;COPY THE LINE BIT MAP.
015670  012700  000206                  MOV     #206,R0         ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
015674  004767  001012                 JSR     PC,WTWLNC       ;INITILAISE THE LINE CONTROL REGISTER.
015700  012700  177670                  MOV     #177670,R0      ;PASS THE LPR CONTENTS.
015704  004767  001032                 JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
015710  012704  000012                  MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
015714  004767  176000                 JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.

015720                          60$:    PASS                            ;RESTORE GPRS.
015720  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
015722  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.
                                ;-- ************************************************************************

015724                          SKPSTS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
015724  004567  166164                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015730  012704  000012                 MOV     #10.,R4                 ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
015734  004767  175760                JSR     PC,DELAY                ;DELAY FOR 10 MILLI-SECONDS.
                                ;+
                                ; Write skip self-test code (52525) to all the indexed DUT Registers.
                                ;-
015740  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.
015744  012703  052525                  MOV     #52525,R3               ;INITIALISE THE SKIP SELF-TEST CODE.
015750  005301                4$:       DEC     R1                      ;SELECT THE NEXT SET OF DEVICE REGISTERS.
015752  016704  164264                  MOV     CSRA,R4                 ;GET THE ADDRESS OF THE CSR OF THE DUT.
015756  010124                          MOV     R1,(R4)+                ;SELECT A BANK OF DUT REGISTERS.
015760  010324                6$:       MOV     R3,(R4)+                ;WRITE THE CODE TO A DUT REGISTER.
015762  020467  164272                  CMP     R4,TXBFCA               ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
015766  103774                         BLO     6$                      ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
015770  032701  000017                  BIT     #17,R1                  ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
015774  001365                         BNE     4$                      ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.

015776                          60$:    PASS                            ;RESTORE GPRS.
015776  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
016000  000207                         RTS     PC
```

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

016002                          TSABRT:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
016002  004567  166106                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016006  012701  016024                 MOV     #2$,R1          ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
016012  012767  011662  166072         MOV     #ER1603,ERRBLK  ;SET-UP THE ERROR REPORTING ROUTINE.
016020                                 ERROR                   ;                >>>>> ERROR <<<<<.
016020  104460                                                                    TRAP    C$ERROR
016022  000432                         BR      60$             ;
016024     040     116     117  2$:    .ASCIZ / NON-RELATED TEST ERROR FOUND DURING TEST EXECUTION/
016027     116     055     122
016032     105     114     101
016035     124     105     104
016040     040     124     105
016043     123     124     040
016046     105     122     122
016051     117     122     040
016054     106     117     125
016057     116     104     040
016062     104     125     122
016065     111     116     107
016070     040     124     105
016073     123     124     040
016076     105     130     105
016101     103     125     124
016104     111     117     116
016107     000
                                .EVEN
016110                          60$:    PASS                     ;RESTORE GPRS.
016110  004736                                         JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
016112  000207                         RTS     PC
```

GLOBAL SUBROUTINE - TXDATP -

```
                                .SBTTL  GLOBAL SUBROUTINE                - TXDATP -
                                ;++ ************************************************************************
                                ;*                        - TRANSMIT DATA PATTERN -
                                ;*      This subroutine is used in the FIHAVL.TST.
                                ;*      It transmits a specified number of data bytes on the specified line.
                                ;*
                                ;* INPUTS:      R0 - Contains the number of data bytes to TX.
                                ;*              R1 - Contains line numb on which transmission is to take place.
                                ;*              BUFBAS to BUFMID contains a 256 byte data pattern.
                                ;*
                                ;* OUTPUTS:     Data is sent out on the specified line.
                                ;*              Carry set = TX successful.
                                ;*
                                ;* CALLING SEQUENCE:    TXDATP
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: DODMA.
                                ;-- ************************************************************************

016114                          TXDATP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016114  004567  165774                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016120  010003                  MOV     R0,R3                           ;PASS THE NUMBER OF CHARS TO TX.
016122  012702  002704          MOV     #BUFBAS,R2                      ;PASS THE START OF THE DATA PATTERN TO TX.
016126  004767  175626          JSR     PC,DODMA                        ;TRANSMIT THE DATA PATTERN.
016132                          60$:    PASS                            ;RESTORE GPRS.
016132  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
016134  000207                  RTS     PC
```

GLOBAL SUBROUTINE - TXDSBL -

```
                                .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 available.
                                ;*               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.
                                ;-- ************************************************************************

016136                          TXDSBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016136  004567  165752                           JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016142  010500                          MOV     R5,R0                   ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
016144  012701  000001                  MOV     #BIT0,R1                ;INITIALIZE THE SELECTED LINE BIT MASK.
016150  016702  164102                  MOV     TXAD2A,R2               ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
016154  005202                          INC     R2                      ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
016156  012703  000010                  MOV     #NUMLNS,R3              ;GET MAXIMUM LINE NUMBER PLUS ONE.
016162  016704  164102                  MOV     IESTAT,R4               ;GET THE STATES OF THE INT ENABLE BITS.
016166  005005                          CLR     R5                      ;LOG POSSIBLE TX DISABLED ON ALL LINES.
                                ;+
                                ; Select every line in turn, and log the state of each TX.ENABLE bit.
                                ;-
016170  010477  164046          2$:     MOV     R4,@CSRA                ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
016174  105712                          TSTB    (R2)                    ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
016176  100001                  BPL     4$                              ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
016200  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.
                                ;-
016202  030100                  4$:     BIT     R1,R0                   ;CHECK STATE OF DISABLE LINE BIT MAP.
016204  001402                  BEQ     6$                              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
016206  142712  000200                  BICB    #BIT7,(R2)              ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
016212  005204                  6$:     INC     R4                      ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
016214  006301                          ASL     R1                      ;SHIFT BIT MAP FOR NEXT LINE.
016216  005303                          DEC     R3                      ;DECREMENT LINE NUMBER.
016220  001363                  BNE     2$                              ;LOOP TO CHECK NEXT LINE.

016222                          60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
016222  010566  000014                                  MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
016226  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                        ;R5 - PREVIOUS STATES OF ALL TX.ENABLE BITS.
016230  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.
                                        ;-- ************************************************************************

016232                                  TXENBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016232  004567  165656                                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016236  010500                                  MOV     R5,R0                   ;COPY BIT MAP OF LINES TO ENABLE.
016240  012701  000001                          MOV     #BIT0,R1                ;INITIALIZE THE SELECTED LINE BIT MASK.
016244  016702  164006                          MOV     TXAD2A,R2               ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
016250  005202                                  INC     R2                      ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
016252  012703  000010                          MOV     #NUMLNS,R3              ;GET MAXIMUM LINE NUMBER.
016256  016704  164006                          MOV     IESTAT,R4               ;GET THE STATES OF THE INT ENABLE BITS.
016262  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.
                                        ;-
016264  010477  163752                  2$:     MOV     R4,@CSRA                ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
016270  105712                          TSTB    (R2)                            ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
016272  100401                          BMI     4$                              ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
016274  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.
                                        ;-
016276  030100                  4$:     BIT     R1,R0                           ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
016300  001402                          BEQ     6$                              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
016302  152712  000200                  BISB    #BIT7,(R2)                      ;ENABLE TRANSMISSION ON SELECTED LINE.
016306  005204                  6$:     INC     R4                              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
016310  006301                          ASL     R1                              ;SHIFT BIT MAP FOR NEXT LINE.
016312  005303                          DEC     R3                              ;DECREMENT LINE NUMBER.
016314  001363                          BNE     2$                              ;LOOP TO CHECK NEXT LINE.

016316                          60$:    PASS    R5                              ;RESTORE GPRS,EXCEPT
016316  010566  000014                                  MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
016322  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                                ;R5 - LINE BIT MAP CORRESPONDING TO THE
                                                                                ; PREVIOUS LINES THAT WERE DISABLED.
016324  000207                          RTS     PC
```

```
                                        .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.
                                        ;-- ************************************************************************
016326  010046                          TXIE0::  MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
016330                                         GETPRI  -(SP)            ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
016330  104440                                                                    TRAP    C$GPRI
016332  010046                                                                    MOV     R0,-(SP)
016334                                         SETPRI  #PRI07           ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
016334  012700  000340                                                           MOV     #PRI07,R0
016340  104441                                                                    TRAP    C$SPRI
016342  042767  177677  163720               BIC     #177677,IESTAT   ;CLEAR TX.INT.ENBL BIT IN IESTAT.
016350  016777  163714  163664               MOV     IESTAT,@CSRA     ;DISABLE TX INTERRUPTS.
016356                                         SETPRI (SP)+             ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
016356  012600                                                           MOV     (SP)+,R0
016360  104441                                                           TRAP    C$SPRI
016362  012600                               MOV     (SP)+,R0         ;RESTORE R0.
016364  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.
                                ;-- ************************************************************************

016366                          UNSDIV:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016366  004567  165522                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;+
                                ; Check for quotient greater than 16 bits condition.
                                ;-
016372  010204                          MOV     R2,R4                   ;GET MSW OF DIVIDEND FOR SUBTRACT.
016374  160104                          SUB     R1,R4                   ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
016376  103403                          BCS     2$                      ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
016400  012701  177777                  MOV     #-1,R1                  ;SET QUOTIENT TO ALL ONES (177777),
016404  000442                          BR      60$                     ;EXIT WITH CARRY CLEAR.
                                ;+
                                ; Set up counters and various working GPRs.
                                ;-
016406  005004                  2$:     CLR     R4                      ;CLEAR THE LSW OF THE DIVISOR.
016410  000241                          CLC                             ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
016412  006001                          ROR     R1                      ; DIVISOR BY
016414  006004                          ROR     R4                      ; 2(UNSIGNED)
016416  012700  000020                  MOV     #16.,R0                 ;SET UP INITIAL SHIFT COUNT TO 16.
                                ;+
                                ; The subtract and shift loop.
                                ;-
016422  010246                  4$:     MOV     R2,-(SP)                ;SAVE MSWORD OF DIVIDEND.
016424  010346                          MOV     R3,-(SP)                ;SAVE LSWORD OF DIVIDEND.
016426  160403                          SUB     R4,R3                   ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
016430  005602                          SBC     R2                      ;MSWORD DIVIDEND - BORROW .
016432  103402                          BCS     6$                      ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
016434  160102                          SUB     R1,R2                   ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
016436  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.
                                    ;-
016440  012603                  6$:     MOV     (SP)+,R3                ;RESTORE LSWORD OF DIVIDEND.
016442  012602                          MOV     (SP)+,R2                ;RESTORE MSWORD OF DIVIDEND.
```

GLOBAL SUBROUTINE - UNSDIV -

```
3967 016444  000401                        BR      10$             ;GOTO SHIFT 1 INTO THE QUOTIENT.
3968                                    ;+
3969                                    ; It went, so we restore the stack and shift a 0 into quotient (will be
3970                                    ; complemented later).   Carry is clear.
3971                                    ;-
3972 016446  012626                 8$:     MOV     (SP)+,(SP)+     ;POP THE SAVED DIVIDEND OFF OF THE STACK.
3973                                    ;+
3974                                    ; Shift the result of the subtract attempt into the quotient shift reg.
3975                                    ;-
3976 016450  006105                 10$:    ROL     R5              ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
3977 016452  000241                         CLC                     ;DIVIDE THE
3978 016454  006001                         ROR     R1              ;  DEVISOR BY
3979 016456  006004                         ROR     R4              ;  2 (UNSIGNED).
3980 016460  005300                         DEC     R0              ;COUNT THIS SHIFT AND SUBTRACT.
3981 016462  001357                         BNE     4$              ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
3982 016464  005105                         COM     R5              ;GET QUOTIENT FROM INVERTED QUOTIENT.
3983                                    ;+
3984                                    ; Now we either round up or leave quotient alone.
3985                                    ;-
3986 016466  000241                         CLC                     ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
3987 016470  006103                         ROL     R3              ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
3988 016472  103402                         BCS     12$             ;IF CARRY FROM SHIFT, ROUND UP.
3989 016474  160403                         SUB     R4,R3           ;SUBTRACT DIVISOR FROM DIVIDEND.
3990 016476  103403                         BCS     14$             ;IF BORROW, DON'T ROUND UP.
3991                                    ;+
3992                                    ; Round up, extra subtract went.
3993                                    ;-
3994 016500  005205                 12$:    INC     R5              ;INCREMENT THE QUOTIENT BY ONE.
3995 016502  001001                         BNE     14$             ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
3996 016504  005305                         DEC     R5              ;DON'T LET ROUNDING CAUSE OVERFLOW.
3997                                    ;+
3998                                    ; All done, pass quotient and exit.
3999                                    ;-
4000 016506  010501                 14$:    MOV     R5,R1           ;PASS QUOTIENT BACK IN R1.
4001 016510  000261                         SEC                     ;INDICATE NO OVERFLOW.
4002
4003 016512                         60$:    PASS    R1                      ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
     016512  010166  000004                                 MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
     016516  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
4004                                                                ;R1 - 16 BIT, UNSIGNED QUOTIENT,
4005 016520  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 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 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.
                        ;-- ************************************************************************

016522                  WAIBIC:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016522  004567  165366                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016526  010204                  MOV     R2,R4                   ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
016530  010102                  MOV     R1,R2
016532  042701  170000          BIC     #170000,R1              ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
016536  042702  007777          BIC     #7777,R2                ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
016542  000302                  SWAB    R2                      ;PUT LINE NUMBER FIELD IN LSBYTE.
016544  006202                  ASR     R2                      ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
016546  006202                  ASR     R2                      ;  POSITION TO USE IT AS A WORD TABLE OFFSET
016550  006202                  ASR     R2                      ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
016552  016202  002370          MOV     BITTBL(R2),R2           ;GET BIT MAP OF LINE TO TEST FROM TABLE.
016556  005003                  CLR     R3                      ;INDICATE THAT THE BIT SHOULD BE CLR.
016560  004767  175610          JSR     PC,MSLGET               ;WAIT FOR THE BIT TO BE CLR WITHIN TIME-OUT.
                                                                ;  CARRY IS CORRECT UPON MSLGET RETURN.
016564  010002                  MOV     R0,R2                   ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
016566                  60$:    PASS    R2                      ;RESTORE GPRS, EXCEPT THE FOLLOWING:
016566  010266  000006                          MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
016572  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
016574  000207                  RTS     PC                      ;  CARRY - SUCCESS FLAG (SET IF BIT FOUND CLR).
```

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

016576                  WAIBIS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016576  004567  165312                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016602  010204                  MOV     R2,R4                   ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
016604  010102                  MOV     R1,R2
016606  042701  170000          BIC     #170000,R1              ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
016612  042702  007777          BIC     #7777,R2                ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
016616  000302                  SWAB    R2                      ;PUT LINE NUMBER FIELD IN LSBYTE.
016620  006202                  ASR     R2                      ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
016622  006202                  ASR     R2                      ;  POSITION TO USE IT AS A WORD TABLE OFFSET
016624  006202                  ASR     R2                      ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
016626  016202  002370          MOV     BITTBL(R2),R2           ;GET BIT MAP OF LINE TO TEST FROM TABLE.
016632  010203                  MOV     R2,R3                   ;INDICATE THAT THE BIT SHOULD BE SET.
016634  004767  175534          JSR     PC,MSLGET               ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
                                                                ;  CARRY IS CORRECT UPON MSLGET RETURN.
016640  010002                  MOV     R0,R2                   ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
016642                  60$:    PASS    R2                      ;RESTORE GPRS, EXCEPT THE FOLLOWING:
016642  010266  000006                          MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
016646  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
016650  000207                  RTS     PC                      ;  CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - WAITTX -
                        ;** ************************************************************************
                        ;*                          - WAIT FOR TX TO FINISH -
                        ;*       This subroutine is used in the FIHAVL.TST.
                        ;*      It waits for transmission to complete ie TX_ACTION. Then delays
                        ;*      for 5 milliseconds to allow time for the last character to get into
                        ;*      the fifo.
                        ;*
                        ;* INPUTS:      CSRA - Contains the address of the CSR.
                        ;*
                        ;* OUTPUTS:     Carry - Set indicates success.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WAITTX
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: DELAY,WAIBIS.
                        ;-- ************************************************************************

016652                  WAITTX:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016652  004567  165236                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016656  012701  170536          MOV     #170536,R1      ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
016662  016702  163354          MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
016666  004767  177704          JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
016672  103005                  BCC     60$             ;BRANCH IF FIFO EMPTY, ABORT THE TEST.
016674  012704  000005          MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
016700  004767  175014          JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
016704  000261                  SEC                     ;SET CARRY TO I8NDICATE SUCCESS.

016706                  60$:    PASS                    ;RESTORE GPRS.
016706  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                ;PASS THE CARRY BIT, SET INDICATES SUCCESS.
016710  000207                  RTS     PC
```

```
                                .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 bits 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.
                                ;-- ************************************************************************

016712                          WTWLNC:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016712  004567  165176                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;+
                                ; Set up the parameters for the call to ALTFLD.
                                ;-
016716  016701  163330                  MOV     LNCTRA,R1               ;SET UP THE REGISTER ADDRESS PARAMETER.
016722  010002                          MOV     R0,R2                   ;SET UP THE DESIRED REGISTER CONTENTS.
016724  010503                          MOV     R5,R3                   ;SET UP THE BIT MAP OF LINES TO ALTER.
016726  012704  177777                  MOV     #-1,R4                  ;SELECT ALL REGISTER BITS TO BE ALTERED.
                                ;+
                                ; Call the subroutine which alters the register contents.
                                ;-
016732  004767  174042                  JSR     PC,ALTFLD               ;ALTER THE REGISTER CONTENTS.

016736                          60$:    PASS                            ;RESTORE GPRS.
016736  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
016740  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.
                                ;-- ************************************************************************

016742                          WTWLPR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016742  004567  165146                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;+
                                ; Set up the parameters for the call to ALTFLD.
                                ;-
016746  016701  163274                  MOV     LPRA,R1         ;SET UP THE REGISTER ADDRESS PARAMETER.
016752  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
016754  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
016756  012704  177777          MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                                ;+
                                ; Call the subroutine which alters the register contents.
                                ;-
016762  004767  174012          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

016766                          60$:    PASS                            ;RESTORE GPRS.
016766  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
016770  000207                  RTS     PC
```

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

016772  005767  163330        CLKINT:: TST     TIMER1          ;CHECK FOR TIMER1 AT ZERO.
016776  001402                         BEQ     2$              ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
017000  005367  163322                 DEC     TIMER1          ;DECREMENT TIME COUNT.
017004  005767  163320        2$:      TST     TIMER2          ;CHECK FOR TIMER2 AT ZERO.
017010  001402                         BEQ     4$              ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
017012  005367  163312                 DEC     TIMER2          ;DECREMENT TIME COUNT.
017016  005367  163310          4$:    DEC     TIMER3          ;DECREMENT THE BREAK COUNT.
017022  001006                         BNE     60$             ;EXIT IF NOT TIME TO CALL BREAK.
017024  016767  163304  163300        MOV     BCOUNT,TIMER3   ;SET UP TIME TILL NEXT BREAK.
017032  010046                         MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 FROM BREAK MACRO.
017034                                 BREAK                   ;CHECK FOR OPERATOR CONTROL/C.
017034  104422                                                         TRAP    C$BRK
017036  012600                         MOV     (SP)+,R0        ;RESTORE CONTENTS OF R0.
017040  000002                 60$:    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.
                                ;*************************************************************************

017042  021627  013470          TP4RTN:: CMP    (SP),#ADRPTR    ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
017046  001402                          BEQ     2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
017050  000177  163234                  JMP     @TP4VEC         ;IF NOT,JUMP TO NORMAL 004 TRAP SERVICE RTN.
017054  052767  100000  163230  2$:     BIS     #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
017062  000002                          RTI                     ;ALL DONE, GO BACK TO THE TEST.
```

I9


```

                                   ;*******************************************************************************
**
                                   ;
                                   ;           VDHB.RPT
                                   ;
                                   ;*******************************************************************************



                                   .SBTTL  REPORT CODING SECTION

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

017064                                     BGNRPT
017064                                                                         L$RPT::

017064                                     EXIT    RPT
017064  000167                                                                         .WORD   J$JMP
017066  000000                                                                         .WORD   L10014-2-.

                                           .EVEN

017070                                     ENDRPT
017070                                                                         L10014:
017070  104425                                                                         TRAP    C$RPT
```

```
                                        .SBTTL  PROTECTION TABLE

                                        ;************************************************************************************
*
                                        ;
                                        ;               SKL4.P11
                                        ;
                                        ;****************************************************************************



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

     017072                                     BGNPROT
     017072                                                                          L$PROT::

     017072  177777                             -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
     017074  177777                             -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
     017076  177777                             -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

     017100                                     ENDPROT

```

PROTECTION TABLE

```
                                ;*****************************************************************************************
                                ;
                                ;               VDHB.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.
                                ;*
                                ;*************************************************************************
                                ;--
017100                                  BGNINIT
017100                                                                  L$INIT::
                                ;SEE IF PROGRAM JUST STARTED, BR IF YES
017100                                  READEF  #EF.START
017100  012700  000040                                                  MOV     #EF.START,R0
017104  104447                                                         TRAP    C$REFG
017106                                  BCOMPLETE       NEWSTA
017106  103416                                                         BCS     NEWSTA
                                ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
017110                                  READEF  #EF.RESTART
017110  012700  000037                                                  MOV     #EF.RESTART,R0
017114  104447                                                         TRAP    C$REFG
017116                                  BCOMPLETE       NEWRES
017116  103556                                                         BCS     NEWRES
                                ;SEE IF THIS IS A NEW PASS, BR IF YES
017120                                  READEF  #EF.NEW
017120  012700  000035                                                  MOV     #EF.NEW,R0
017124  104447                                                         TRAP    C$REFG
017126                                  BCOMPLETE       NEWPAS
017126  103555                                                         BCS     NEWPAS
                                ;SEE IF PROGRAM WAS JUST CONTINUED
017130                                  READEF  #EF.CONTINUE
017130  012700  000036                                                  MOV     #EF.CONTINUE,R0
017134  104447                                                         TRAP    C$REFG
017136                                  BNCOMPLETE      GETPRM
017136  103161                                                         BCC     GETPRM
017140  000167  000556                 JMP     ENDIT
017144                          NEWSTA:
017144                                  BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
017144  104433                                                         TRAP    C$RESET
                                ;+
                                ; Set up for Line Time Clock interrupts.
                                ;-
017146                                   CLOCK  L,R1            ;GET THE CLOCK PARAMETERS.
017146  012700  000114                                                  MOV     #'L,R0
017152  104462                                                         TRAP    C$CLCK
```

INITIALIZE SECTION

```
        017154  010001                                                          MOV     R0,R1
  4378  017156  012167  163134                MOV     (R1)+,CLKCSR            ;STORE CLOCK CSR ADDRESS.
  4379  017162  012167  163132                MOV     (R1)+,CLKBRL            ;STORE CLOCK BUS REQ INT LEVEL.
  4380  017166  012167  163130                MOV     (R1)+,CLKVEC            ;STORE CLOCK INTERRUPT VECTOR.
  4381  017172  012167  163126                MOV     (R1)+,CLKHRZ            ;STORE CLOCK FREQUENCY.
  4382  017176  026727  163122  000062        CMP     CLKHRZ,#50.             ;TEST FOR 50HZ LINE FREQUENCY.
  4383  017204  001004                        BNE     2$                      ;BRANCH IF CLOCK IS NOT 50HZ.
  4384  017206  012767  000024  163122        MOV     #20.,MSTICK             ;INDICATE 20MS PER CLOCK TICK.
  4385  017214  000403                        BR      4$
  4386  017216  012767  000021  163112  2$:   MOV     #17.,MSTICK             ;INDICATE 17 MS PER CLOCK TICK.
  4387  017224                        4$:     SETVEC  CLKVEC,#CLKINT,#PRI06   ;INITIALIZE CLOCK INTERRUPT VECTOR.
        017224  012746  000300                                                  MOV     #PRI06,-(SP)
        017230  012746  016772                                                  MOV     #CLKINT,-(SP)
        017234  016746  163062                                                  MOV     CLKVEC,-(SP)
        017240  012746  000003                                                  MOV     #3,-(SP)
        017244  104437                                                          TRAP    C$SVEC
        017246  062706  000010                                                  ADD     #10,SP
  4388  017252  016700  163046                MOV     CLKHRZ,R0               ;INITIALIZE THE BREAK COUNT
  4389  017256  006200                        ASR     R0                      ; TO CAUSE A BREAK
  4390  017260  010067  163050                MOV     R0,BCOUNT               ; EVERY 1/2 SECOND.
  4391  017264                                SETPRI  #PRI05                  ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
        017264  012700  000240                                                  MOV     #PRI05,R0
        017270  104441                                                          TRAP    C$SPRI
  4392                                  ;+
  4393                                  ; Enable the Line Time Clock (LTC) checking to make sure that the CSR
  4394                                  ; is accessable.
  4395                                  ;    First set up to catch any 004 traps which occur:
  4396                                  ;-
  4397  017272  016767  160506  163010        MOV     4,TP4VEC                ;SAVE THE EXISTING 004 TRAP VECTOR.
  4398  017300  012767  017042  160476        MOV     #TP4RTN,4               ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
  4399                                  ;+
  4400                                  ;    Enable LTC checking for 004 trap in case CSR is not there.
  4401                                  ;-
  4402  017306  005067  163000                CLR     TP4FLG                  ;CLEAR THE 004 TRAP FLAG.
  4403  017312  012767  000100  162774        MOV     #BIT6,WORD1             ;SET UP TO SET BIT6 OF THE LTC CSR.
  4404  017320  012700  002314                MOV     #WORD1,R0               ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
  4405  017324  016701  162766                MOV     CLKCSR,R1               ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
  4406  017330  004767  174122                JSR     PC,CKTRAP               ;MOVE AND CHECK FOR TRAP.
  4407  017334  016767  162750  160442        MOV     TP4VEC,4                ;RESTORE THE NORMAL 004 TRAP VECTOR.
  4408  017342  103403                        BCS     6$                      ;IF NO TRAP, LTC IS THERE SO CONTINUE.
  4409  017344  005067  162754                CLR     CLKHRZ                  ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
  4410  017350  000402                        BR      8$                      ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
  4411                                  ;+
  4412                                  ; Calibrate the DELAY routine milli-second delay count value.
  4413                                  ;-
  4414  017352  004767  173604        6$:     JSR     PC,CALMSL
  4415                                  ;+
  4416                                  ; Check for Memmory Management present on this machine.
  4417                                  ; If MEM MGT is present, disable it.
  4418                                  ;-
  4419  017356  016767  160422  162724  8$:   MOV     4,TP4VEC                ;SAVE THE EXISTING 004 TRAP VECTOR.
  4420  017364  012767  017042  160412        MOV     #TP4RTN,4               ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
  4421  017372  005067  162714                CLR     TP4FLG                  ;CLEAR THE 004 TRAP FLAG.
  4422  017376  005067  162712                CLR     WORD1                   ;PREPARE TO CLEAR THE MEM MGT SR0 REGISTER.
  4423  017402  012700  002314                MOV     #WORD1,R0               ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
  4424  017406  016701  162730                MOV     MMSR0,R1                ;SELECT MEM MGT SR0 REGISTER AS DESTINATION.
  4425  017412  005067  162726                CLR     MMPRES                  ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
```

INITIALIZE SECTION

```
4426 017416  005067  162724                    CLR     MMENAB          ;INDICATE MEM MGT IS NOT ENABLED.
4427 017422  004767  174030                    JSR     PC,CKTRAP       ;CLEAR THE MEM MCT SR0 REG AND CHECK FOR TRAP.
4428 017426  016767  162656  160350            MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4429 017434  103003                            BCC     10$             ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
4430 017436  012767  000001  162700            MOV     #1,MMPRES       ;INDICATE THAT MEM MGT IS PRESENT.
4431 017444  005067  162626            10$:    CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4432 017450  000167  000006                    JMP     NEWPAS          ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
4433
4434 017454                            NEWRES: BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
     017454  104433                                                    TRAP    C$RESET
4435 017456  005067  162614                    CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4436 017462                            NEWPAS:
4437 017462  012767  177777  162550            MOV     #-1,UNITN       ;RESET LOGICAL DEVICE TO -1
4438                                   ;+
4439                                   ; Increment the pass counter, correct for any overflow.
4440                                   ;  This counter is used in the Rom version test.
4441                                   ;-
4442 017470  005267  162602                    INC     PASCNT          ;INCREMENT THE PASS COUNTER.
4443 017474  001002                            BNE     GETPRM          ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
4444 017476  005367  162574                    DEC     PASCNT          ;SET PASS COUNT TO 177777 OCTAL.
4445
4446                                   ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
4447 017502                            GETPRM:
4448 017502  005267  162532                    INC     UNITN           ;INCREMENT LOGICAL DEVICE NUMBER
4449 017506  026767  162526  162276            CMP     UNITN,L$UNIT    ;SEE IF MAXIMUM UNIT NO. EXCEEDED
4450 017514  002362                            BGE     NEWPAS          ;BR IF YES
4451
4452 017516                                    GPHARD  UNITN,R1        ;GET P-TABLE POINTER INTO R1
     017516  016700  162516                                                    MOV     UNITN,R0
     017522  104442                                                            TRAP    C$GPHRD
     017524  010001                                                            MOV     R0,R1
4453 017526                                    BCOMPLETE       30$     ;BR IF DEVICE AVAILABLE
     017526  103401                                                            BCS     30$
4454 017530  000764                            BR      GETPRM          ;SKIP THIS DEVICE
4455
4456
4457                                   ;*********** HARDWARE PARAMETER MOVING CODE ****************
4458 017532  012167  162504    30$:     MOV     (R1)+,CSRA      ;STORE DHV11-M CSR ADDRESS IN DEV.REG.ADDRESS TABLE
4459 017536  012102                     MOV     (R1)+,R2        ;GET THE RX INTERRUPT VECTOR ADDRESS.
4460 017540  010267  162464             MOV     R2,RXVECA       ;STORE RX INT VECTOR ADDRESS.
4461 017544  062702  000004             ADD     #4,R2           ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
4462 017550  010267  162456             MOV     R2,TXVECA       ;STORE TX INT VECTOR ADDRESS.
4463 017554  012167  162454             MOV     (R1)+,ACTLNS    ;STORE DHV11-M ACTIVE LINE BIT MAP
4464 017560  012702  000377             MOV     #MAPLNS,R2      ;GET THE BIT MAP FOR ALL LINES.
4465 017564  005102                     COM     R2              ;GET A BIT MAP OF NON-EXISTANT LINES.
4466 017566  040267  162442             BIC     R2,ACTLNS       ;CLEAR NON-EXISTANT LINES FROM ACTLNS.
4467 017572  112167  162440             MOVB    (R1)+,LOPBCK    ;STORE DHV11-M LOOPBACK MODE
4468 017576  112167  162435             MOVB    (R1)+,BRLEVL    ;STORE DHV11-M INTERUPT BUS REQUEST LEVEL
4469                                   ;+
4470                                   ;       CALCULTATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
4471                                   ;       DEVICE REGISTER ADDRESS TABLE.
4472                                   ;-
4473 017602  016701  162434             MOV     CSRA,R1         ;COPY CSR ADDRESS
4474 017606  005201                     INC     R1              ;INCREMENT CSR ADDRESS
4475 017610  005201                     INC     R1              ; COPY BY 2.
4476 017612  012703  000007             MOV     #7,R3           ;SET UP REGISTER COUNT
4477 017616  012702  002244             MOV     #RBUFA,R2       ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
```

INITIALIZE SECTION

```
 4478 017622  010122                 12$:    MOV     R1,(R2)+        ;STORE REGISTER ADDRESS IN TABLE
 4479 017624  005201                         INC     R1              ;INCREMENT REGISTER ADDRESS
 4480 017626  005201                         INC     R1              ; BY 2,FOR THE NEXT DEVICE REGISTER.
 4481 017630  005303                         DEC     R3              ;DECREMENT REGISTER COUNT
 4482 017632  001373                       BNE     12$               ;LOOP IF NOT DONE
 4483
 4484                                     ;+
 4485                                     ; Initialise the BMP code queue.
 4486                                     ;-
 4487 017634  012700  002444                 MOV     #BMPCQB,R0      ;GET THE START ADDRESS OF THE QUEUE.
 4488 017640  012701  002644                 MOV     #BMPCQE,R1      ;GET THE END ADDRESS OF THE QUEUE.
 4489 017644  010067  162572                 MOV     R0,BMPCQP       ;SET THE POINTER TO THE START OF THE QUEUE.
 4490 017650  005020                 14$:    CLR     (R0)+           ;CLEAR OUT THE CONTENTS OF THE QUEUE.
 4491 017652  020001                         CMP     R0,R1           ;CHECK IF END OF QUEUE HAS BEEN REACHED.
 4492 017654  103775                       BLO     14$               ;LOOP IF NOT ALL DONE.
 4493                                     ;+
 4494                                     ; Report the Unit number if the software P-table question was answered YES,
 4495                                     ;  and the maximum unit number is greater than 1.
 4496                                     ;-
 4497 017656  032767  000020  162340         BIT     #BIT4,OPTION    ;CHECK IF THE QUESTION WAS ANSWERED YES.
 4498 017664  001416                       BEQ     16$               ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
 4499 017666  026727  162120  000001         CMP     L$UNIT,#1       ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
 4500 017674  003412                       BLE     16$               ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
 4501 017676                               PRINTF  #MFUNIT,UNITN     ;REPORT UNIT NUMBER.
      017676  016746  162336                                                 MOV     UNITN,-(SP)
      017702  012746  004216                                                 MOV     #MFUNIT,-(SP)
      017706  012746  000002                                                 MOV     #2,-(SP)
      017712  010600                                                         MOV     SP,R0
      017714  104417                                                         TRAP    C$PNTF
      017716  062706  000006                                                 ADD     #6,SP
 4502 017722                         16$:
 4503
 4504 017722  005067  162336         ENDIT:  CLR     CTRLCF          ;CLR THE CTRL-C TEST ABORT FLAG.
 4505                                     ;+
 4506                                     ; Set the processor priority to allow LTC interrupts but not others.
 4507                                     ;-
 4508 017726                               SETPRI  #PRI05                  ;SET PROCESSOR PRIORITY TO 5.
      017726  012700  000240                                                 MOV     #PRI05,R0
      017732  104441                                                         TRAP    C$SPRI
 4509                                     ;+
 4510                                     ;    Enable Line Time Clock if one is available.
 4511                                     ;-
 4512 017734  005767  162364                 TST     CLKHRZ          ;CHECK FOR A LTC BEING PRESENT.
 4513 017740  001403                       BEQ     18$               ;LTC PRESENT?  NO, SKIP LTC ENABLE.
 4514 017742  012777  000100  162346         MOV     #BIT6,@CLKCSR   ;YES, ENABLE THE LTC.
 4515 017750                         18$:
 4516
 4517 017750                               ENDINIT
      017750                                                                 L10016:
      017750  104411                                                         TRAP    C$INIT
 4518
 4519         000000                       TNUM == 0                  ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.
```

INITIALIZE SECTION

```
                        ;*****************************************************************************
                        ;
                        ;               VDHB.ATD
                        ;
                        ;*****************************************************************************


                        .SBTTL  AUTODROP SECTION


                        ;++
                        ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
                        ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
                        ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
                        ; DROPPED FROM TESTING.
                        ;--

017752                          BGNAUTO
017752                                                          L$AUTO::

017752                          ENDAUTO
017752                                                          L10017:
017752  104461                                                          TRAP    C$AUTO
```

AUTODROP SECTION

```
                                ;*****************************************************************************
                                ;
                                ;               VDHB.CUC
                                ;
                                ;*****************************************************************************



                                .SBTTL  CLEANUP CODING SECTION

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

                                        BGNCLN
                                                                                        L$CLEAN::

                                         TST    CTRLCF          ;DID WE GET HERE BY CTRL-C FROM TEST?
                                        BEQ     2$              ;CTRL-C FROM TEST?  NO, SKIP BUS RESET.
                                        BRESET                  ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
017762  104433                                                                    TRAP    C$RESET
                                2$:     SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS, DISALLOW OTHERS.
017764  012700  000240                                                          MOV     #PRI05,R0
017770  104441                                                                  TRAP    C$SPRI
                                         TST    CLKHRZ          ;SEE IF THE CLOCK IS ON
                                        BEQ     3$              ;IF NOT GO ON
                                        CLR     @CLKCSR         ;STOP THE CLOCK
                                3$:


                                        EXIT    CLN
020004  104432                                                                  TRAP    C$EXIT
020006  000002                                                                  .WORD   L10020-.


                                        .EVEN

                                        ENDCLN
                                                                                L10020:
020010  104412                                                                  TRAP    C$CLEAN
```

CLEANUP CODING SECTION

```
                                ;*****************************************************************************************
                                ;
                                ;               VDHB.DRP
                                ;
                                ;*****************************************************************************************


                                .SBTTL  DROP UNIT SECTION

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

020012                                  BGNDU
020012                                                                  L$DU::

                                        PRINTF  #DROP,R0        ;REPORT UNIT THAT HAS BEEN DROPPED.
020012  010046                                                          MOV     R0,-(SP)
020014  012746  020040                                                  MOV     #DROP,-(SP)
020020  012746  000002                                                  MOV     #2,-(SP)
020024  010600                                                          MOV     SP,R0
020026  104417                                                          TRAP    C$PNTF
020030  062706  000006                                                  ADD     #6,SP

020034                                  EXIT    DU
020034  000167                                                          .WORD   J$JMP
020036  000056                                                          .WORD   L10021-2-.

020040    045   101   040   DROP:   .ASCIZ/%A UNIT%D6%A DROPPED FROM FURTHER TESTING.%N/
020043    125   116   111
020046    124   045   104
020051    066   045   101
020054    040   104   122
020057    117   120   120
020062    105   104   040
020065    106   122   117
020070    115   040   106
020073    125   122   124
020076    110   105   122
020101    040   124   105
020104    123   124   111
020107    116   107   056
020112    045   116   000

                                .EVEN

                                ENDDU
020116                                                          L10021:
020116  104453                                                          TRAP    C$DU
```

DROP UNIT SECTION

```
                                ;*****************************************************************************
                                ;
                                ;               VDHB.ADD
                                ;
                                ;*****************************************************************************


                                .SBTTL  ADD UNIT SECTION

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

020120                                  BGNAU
020120                                                                  L$AU::


                                        EXIT    AU
020120  000167                                                                  .WORD   J$JMP
020122  000000                                                                  .WORD   L10022-2-.


                                .EVEN

                                ENDAU
020124
020124                                                                  L10022:
020124  104452                                                                  TRAP    C$AU
```

HARDWARE TEST - ADRA -

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

020126                                  BGNTST
020126                                                                          T1::
000001                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
020126  012767  000001  162132   MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (1)
020134  012767  177777  162122   MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.

                                ;+
                                ; Set up to catch any 004 traps which occur:
                                ;-
020142  016767  157636  162140   MOV    4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
020150  012767  017042  157626   MOV    #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
020156  005005                   CLR    R5              ;CLEAR THE ERROR FLAGS.

                                ;+
                                ; Set up for the initial iteration of the test loop:
                                ;-
020160  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.
                                ;-
020162  005067  162124          2$:     CLR    TP4FLG          ;CLEAR THE 004 TRAP FLAG.
020166  016700  162050           MOV    CSRA,R0         ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
020172  012701  020406           MOV    #52$,R1         ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
020176  004767  173254          JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
020202  103402                  BCS     4$              ;IF NO TRAP, BYPASS ERROR.
020204  052705  100001           BIS    #100001,R5      ;SET FATAL READ ERROR FLAGS.
020210  042767  000017  000170  4$:     BIC    #17,52$         ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
020216  050467  000164           BIS    R4,52$          ;OR IN THE LINE COUNTER TO THE I.A.R FIELD.
020222  010100                   MOV    R1,R0           ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
020224  016701  162012           MOV    CSRA,R1         ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
020230  004767  173222          JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
020234  103403                  BCS     6$              ;IF NO TRAP, BYPASS ERROR.
020236  052705  100002           BIS    #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
020242  000440                  BR      40$             ;EXIT AND REPORT FATAL ERROR.

                                ;+
                                ;    Now, we test each register for this line.
                                ;-
020244  012702  000010          6$:     MOV    #10,R2          ;INIT REGISTER COUNTER TO 8.
020250  016767  161766  000126   MOV    CSRA,50$                ;INITIALIZE THE REGISTER POINTER.
020256  012700  020404          8$:     MOV    #50$,R0         ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
020262  012701  020406           MOV    #52$,R1         ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
020266  004767  173164          JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
```

```
 4732 020272  103402                    BCS     10$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
 4733 020274  052705  100001             BIS     #100001,R5      ;SET FATAL READ ERROR FLAGS.
 4734 020300  010100              10$:   MOV     R1,R0           ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
 4735 020302  012701  020404             MOV     #50$,R1         ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.
 4736 020306  004767  173144             JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
 4737 020312  103402                    BCS     12$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
 4738 020314  052705  100002             BIS     #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
 4739 020320  005267  000060       12$:  INC     50$             ;INCREMENT THE REGISTER
 4740 020324  005267  000054             INC     50$             ; POINTER BY 2.
 4741 020330  005302                    DEC     R2              ;COUNT THE REGISTER.
 4742 020332  001351                    BNE     8$              ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
 4743
 4744                              ;+
 4745                              ; Now we set up to test the next line, or to exit if we are done.
 4746                              ;-
 4747 020334  005204                    INC     R4              ;INCREMENT THE LINE COUNTER.
 4748 020336  020427  000010             CMP     R4,#NUMLNS      ;COMPARE LINE COUNTER AGAINST NUMBER OF LINES.
 4749 020342  002707                    BLT     2$              ;LOOP TO TEST THE NEXT LINE IF WE'RE NOT DONE.
 4750
 4751                              ;+
 4752                              ; Done checking device register addresses.
 4753                              ;    Report any errors and exit.
 4754                              ;-
 4755 020344  016767  161740  157432 40$:  MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
 4756 020352  005705                    TST     R5              ;CHECK THE ERROR FLAGS.
 4757 020354  100015                    BPL     60$             ;EXIT ROUTINE IF NO ERRORS.
 4758                                 ; REPORT "DEVICE REGISTER ACCESS ERRORS"
 4759 020356                            ERRDF   101,EM0103,ER0101;              >>>>> ERROR #101 <<<<<.
      020356  104455                                                                  TRAP    C$ERDF
      020360  000145                                                                  .WORD   101
      020362  005473                                                                  .WORD   EM0103
      020364  011320                                                                  .WORD   ER0101
 4760
 4761 020366                            DODU    UNITN           ;DROP THIS UNIT FROM FUTHER TESTING.
      020366  016700  161646                                                          MOV     UNITN,R0
      020372  104451                                                                  TRAP    C$DODU
 4762 020374  005067  161664             CLR     CTRLCF          ;INDICATE NO CTRL-C ABORT FROM TEST.
 4763 020400                            DOCLN                   ;ABORT THIS SUB PASS.
      020400  104444                                                                  TRAP    C$DCLN
 4764 020402  000402                    BR      60$             ;
 4765                              ;+
 4766                              ; Local storage.
 4767                              ;-
 4768 020404  000000              50$:  .WORD   0       ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 4769 020406  000000              52$:  .WORD   0       ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 4770 020410  005067  161650       60$:  CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 4771 020414                            ENDTST
      020414                                                                L10023:
      020414  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - DMASTA -
                                ;++ ******************************************************************************
                                ;*                          - DMA Start Bit Test -
                                ;*      This test verifies that the DMA_START bit in the DUT's Line control
                                ;*      registers will initiate DMA transmission on the selected line.
                                ;*      This test is performed in internal loopback, on all active lines.
                                ;*
                                ;-- ******************************************************************************
020416                                  BGNTST
020416                                                                          T2::
        000002                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
020416  012767  000002  161642           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (40)
020424  012767  177777  161632           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
020432  012767  000001  163444           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
020440  012767  007641  163440           MOV    #4001.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
020446  012767  005774  163434           MOV    #EM4001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
020454  012767  012572  163430           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 >>>>> 4001 <<<<<.
                                ;-
020462  004767  173020                   JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
020466  103141                           BCC     50$             ;RESET FAILURE?, ABORT THIS TEST.

020470  004767  173520                   JSR     PC,INDATP       ;INITIALSE THE 256 BYTE DATA PATTERN.
                                ;+
                                ; Set internal loopback,enable receiver functions 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 active lines.
                                ;-
020474  016705  161534                   MOV    ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
020500  012700  000204                   MOV    #204,R0         ;PASS THE LNCTRL CONTENTS.
020504  004767  176202                   JSR    PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
020510  012700  177670                   MOV    #177670,R0      ;PASS THE LPR CONTENTS.
020514  004767  176222                   JSR    PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
020520  004767  175506                   JSR    PC,TXENBL       ;ENABLE TRANSMITTERS ON ALL LINES.
                                ;+
                                ; Set-up outer loop to test the DMA_START bit on all active lines.
                                ;-
020524  016705  161504                   MOV    ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
020530  005001                           CLR    R1              ;CLEAR THE LINE NUMBER COUNTER.
020532  012767  007642  163346  2$:      MOV    #4002.,ERRNBR   ;SET THE ERROR NUMBER TO 4002.
020540  000241                           CLC                    ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
020542  006005                           ROR    R5              ;SHIFT THE BIT MAP INTO THE CARRY BIT.
020544  103106                          BCC     14$             ;DO NOT TEST THE LINE IF IT IS INACTIVE.
020546  004767  174262                  JSR     PC,PUFIFO       ;PURGE THE FIFO.
020552  103107                          BCC     50$             ;GO REPORT ERROR IF FIFO WILL NOT CLEAR.
                                    ;+
                                    ; Perform DMA_START bit testing on each line individually.
                                    ; Test each DMA_START bit before TX'ing data pattern, report error if set.
                                    ; Set DMA_START bit on LUT, verify it is set, report error if clear.
                                    ; Wait for DMA to complete.
                                    ; Verify DMA_START bit is clear, report error if set.
                                    ; Verify correct number of chars were received, report error if < expected.
                                    ;-
```

HARDWARE TEST - DMASTA -

```
 4829 020554  005267  163326                  INC     ERRNBR          ;SET ERROR NUMBER TO 4003.
 4830 020560  012702  002704                  MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
 4831 020564  012703  000144                  MOV     #100.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
 4832 020570  004767  173164                 JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
 4833 020574  103067                          BCC     12$             ;GO REPORT ERROR IF DMA_START BIT SET.
 4834                                        ;+
 4835                                        ; Test the state of the DMA_START bit on the line under test.
 4836                                        ;  Report error if DMA_START bit is clear.
 4837                                        ;-
 4838 020576  005267  163304                  INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4004.
 4839 020602  010177  161434                  MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
 4840 020606  105777  161444                  TSTB    @TXAD2A         ;TEST THE STATE OF THE DMA_START BIT.
 4841 020612  100060                         BPL     12$             ;GO REPORT ERROR IF BIT IS CLEAR.
 4842                                        ;+
 4843                                        ; Wait for DMA transmission to complete.
 4844                                        ;-
 4845 020614  005267  163266          4$:     INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4005.
 4846 020620  010103                          MOV     R1,R3           ;SAVE THE LINE NUMBER.
 4847 020622  012701  170226                  MOV     #170226,R1      ;TEST BIT 15, TIMEOUT OF 150 MILLI SECS.
 4848 020626  016702  161410                  MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 4849 020632  004767  175740                 JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE.
 4850 020636  103045                         BCC     10$             ;GO REPORT ERROR IF TIMEOUT OCCURRED.
 4851 020640  012704  000005                  MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
 4852 020644  004767  173050                 JSR     PC,DELAY        ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
 4853 020650  010301                          MOV     R3,R1           ;RESTORE THE CURRENT LINE NUMBER.
 4854                                        ;+
 4855                                        ; Test the state of the DMA_START bit on the line under test.
 4856                                        ;  Report error if DMA_START bit is set.
 4857                                        ;-
 4858 020652  005267  163230                  INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4006.
 4859 020656  010177  161360                  MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
 4860 020662  105777  161370                  TSTB    @TXAD2A         ;TEST THE STATE OF THE DMA_START BIT.
 4861 020666  100432                         BMI     12$             ;GO REPORT ERROR IF BIT IS STILL SET.
 4862                                        ;+
 4863                                        ; Verify the number of chars received = number of chars expected.
 4864                                        ; Report error if count is incorrect.
 4865                                        ; If more than 128 BMP codes are found then report error and exit test.
 4866                                        ;-
 4867 020670  005003                          CLR     R3              ;CLEAR THE READ COUNTER.
 4868 020672  012704  000200                  MOV     #128.,R4        ;SET UP MAX BMP CODE READ COUNT.
 4869 020676  012767  007647  163202  6$:     MOV     #4007.,ERRNBR   ;SET ERROR NUMBER TO 4007.
 4870 020704  017702  161334                  MOV     @RBUFA,R2       ;READ THE CHARACTER FROM THE FIFO.
 4871 020710  100021                         BPL     12$             ;GO REPORT ERROR IF FIFO EMPTY TOO SOON.
 4872 020712  012700  170301                  MOV     #170301,R0      ;SET-UP BIT MASK OF A BMP CODE.
 4873 020716  040200                          BIC     R2,R0           ;TRY TO CLEAR THE BMP CODE MASK.
 4874 020720  001007                         BNE     8$              ;BRANCH IF NOT A BMP CODE.
 4875 020722  005267  163160                  INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4008.
 4876 020726  004767  174612                 JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
 4877 020732  005304                          DEC     R4              ;DECREMENT MAX BMP CODE READ COUNT.
 4878 020734  001416                         BEQ     50$             ;GO REPORT ERROR IF TOO MANY BMP CODES FOUND.
 4879 020736  000757                         BR      6$              ;DO NOT COUNT THE BMP CODE AS A VALID CHAR.
 4880 020740  005203                  8$:     INC     R3              ;COUNT THIS CHARACTER.
 4881 020742  020327  000144                  CMP     R3,#100.        ;HAVE WE RECIEVED 100 CHARACTERS?.
 4882 020746  002753                         BLT     6$              ;LOOP UNTIL 100 (NON-BMP) CHARS ARE READ.
 4883 020750  000404                         BR      14$             ;SKIP AROUND THE ERROR REPORT.
 4884
 4885                                       ;+
```

```
                                        ; Report error, skip further testing on this line.
                                        ;-
020752  010301              10$:    MOV     R3,R1           ;RESTORE THE CURRENT LINE NUMBER.

020754  012702  006017      12$:    MOV     #EM4002,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
                                                            ; "DMA_START BIT BAD ON LINE nn".
020760                              ERROR                   ;               >>>>> ERROR <<<<<.
020760  104460                                                                  TRAP    C$ERROR

020762  005201              14$:    INC     R1              ;INCREMENT THE LINE NUMBER COUNTER.
020764  005705                      TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
020766  001261                      BNE     2$              ;YES; BRANCH TO TEST THE NEXT LINE.
020770  000402                      BR      60$             ;NO; EXIT THIS TEST.

020772  004767  175004      50$:    JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
020776  005067  161262      60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.

021002                              ENDTST
021002                                                                  L10024:
021002  104401                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - DMABRT -
                                ;++ ******************************************************************************
                                ;*                          - DMA Abort/Restart Test -
                                ;*      This test verifies that each DMA_ABORT bit will correctly halt
                                ;*      a DMA transmission, and return a TX_ACTION.
                                ;*      It will also verify that the aborted DMA transmission can be resummed,
                                ;*      and that a TX_ACTION is returned upon completion.
                                ;*      This test is performed in internal loopback, on all active lines.
                                ;*
                                ;-- ******************************************************************************
021004                                  BGNTST
021004                                                                          T3::
        000003                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
021004  012767  000003  161254           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (41)
021012  012767  177777  161244           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
021020  012767  000001  163056           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
021026  012767  010005  163052           MOV    #4101.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
021034  012767  006053  163046           MOV    #EM4101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
021042  012767  012572  163042           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 >>>>> 4101 <<<<<.
                                ;-
021050  004767  172432                   JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
021054  103160                           BCC     60$             ;RESET FAILURE?, ABORT THIS TEST.

021056  004767  173132                   JSR     PC,INDATP       ;INITIALISE 256 BYTE DATA PATTERN.
                                ;+
                                ; Set internal loopback,enable receiver functions 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 active lines.
                                ;-
021062  016705  161146                    MOV    ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
021066  012700  000204                    MOV    #204,R0         ;PASS THE LNCTRL CONTENTS.
021072  004767  175614                   JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
021076  012700  177670                    MOV    #177670,R0      ;PASS THE LPR CONTENTS.
021102  004767  175634                   JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
021106  004767  175120                   JSR     PC,TXENBL       ;ENABLE TRANSMITTERS ON ALL LINES.
                                ;+
                                ; Perform DMA_ABORT bit testing on each individual (active) line.
                                ;-
021112  016705  161116                    MOV    ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
021116  005001                            CLR    R1              ;CLEAR THE LINE NUMBER COUNTER.
021120  012767  010006  162760  2$:       MOV    #4102.,ERRNBR   ;SET THE ERROR NUMBER TO 4102.
021126  000241                            CLC                    ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
021130  006005                            ROR    R5              ;SHIFT THE BIT MAP INTO THE CARRY BIT.
021132  103123                           BCC     10$             ;DO NOT TEST THE LINE IF IT IS INACTIVE.
021134  004767  173674                   JSR     PC,PUFIFO       ;PURGE THE FIFO.
021140  103124                           BCC     50$             ;GO REPORT ERROR IF FIFO WILL NOT CLEAR.
                                    ;+
                                    ;  Check the DMA_ABORT bit before enabling DMA, report error if set.
                                    ;-
021142  005267  162740                    INC    ERRNBR          ;INCREMENT ERROR NUMBER TO 4103.
021146  010177  161070                    MOV    R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
021152  032777  000001  161072            BIT    #BIT0,@LNCTRA   ;TEST THE STATE OF THE DMA_ABORT BIT.
```

```
 4960 021160  001105                        BNE     6$              ;GO REPORT ERROR IF BIT IS SET.
 4961                                      ;+
 4962                                      ; Enable DMA TX on selected line, wait for DMA to TX approx 1/4 of data.
 4963                                      ;  Abort the DMA transmission. Wait for TX_ACTION to be returned.
 4964                                      ;-
 4965 021162  005267  162720                 INC     ERRNBR          ;SET ERROR NUMBER TO 4104.
 4966 021166  012702  002704                 MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
 4967 021172  012703  000400                 MOV     #256.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
 4968 021176  004767  172556                 JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
 4969 021202  103103                         BCC     50$             ;GO REPORT ERROR IF THERE ARE TX PROBLEMS.
 4970                                      ;+
 4971                                      ; Wait for DMA to transmit 1/4 of the data before aborting.
 4972                                      ;-
 4973 021204  010177  161032                 MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
 4974 021210  012704  000062                 MOV     #50.,R4         ;PASS THE DELAY TIME OF 40 MILLI SECONDS.
 4975 021214  004767  172500                 JSR     PC,DELAY        ;WAIT FOR APPROX 1/4 OF DATA TO BE TX'D.
 4976 021220  052777  000001  161024         BIS     #BIT0,@LNCTRA   ;ABORT THE DMA TRANSMISSION.
 4977                                      ;+
 4978                                      ; Wait for TX_ACTION to be returned, report error if time-out occurs.
 4979                                      ;-
 4980 021226  005267  162654                 INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4105.
 4981 021232  010103                         MOV     R1,R3           ;SAVE THE LINE NUMBER.
 4982 021234  012701  170012                 MOV     #170012,R1      ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
 4983 021240  016702  160776                 MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 4984 021244  004767  175326                 JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE.
 4985 021250  103050                         BCC     4$              ;GO REPORT ERROR IF TIMEOUT OCCURRED.
 4986 021252  010301                         MOV     R3,R1           ;RESTORE THE CURRENT LINE NUMBER.
 4987                                      ;+
 4988                                      ; Verify DMA_START bit clear, report error if set.
 4989                                      ;-
 4990 021254  005267  162626                 INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4106.
 4991 021260  012702  006132                 MOV     #EM4103,R2      ;SELECT MESSAGE TO BE REPORTED.
 4992                                                                 ; "DMA_START BIT FOUND SET AFTER DMA ABORTED".
 4993 021264  010177  160752                 MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
 4994 021270  105777  160762                 TSTB    @TXAD2A         ;TEST THE STATE OF THE DMA_START BIT.
 4995 021274  100441                         BMI     8$              ;GO REPORT ERROR IF IT IS SET.
 4996                                      ;+
 4997                                      ; Resume DMA transmission by clearing DMA_ABORT and setting DMA_START.
 4998                                      ;-
 4999 021276  042777  000001  160746         BIC     #BIT0,@LNCTRA   ;CLEAR THE DMA_ABORT BIT.
 5000 021304  052777  000200  160744         BIS     #BIT7,@TXAD2A   ;SET THE DMA_START BIT.
 5001                                      ;+
 5002                                      ; Wait for DMA transmission to complete.
 5003                                      ;-
 5004 021312  005267  162570                 INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4107.
 5005 021316  010103                         MOV     R1,R3           ;SAVE THE LINE NUMBER.
 5006 021320  012701  170536                 MOV     #170536,R1      ;TEST BIT 15, TIMEOUT OF 350 MILLI SECS.
 5007 021324  016702  160712                 MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 5008 021330  004767  175242                 JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE.
 5009 021334  103016                         BCC     4$              ;GO REPORT ERROR IF TIMEOUT OCCURRED.
 5010 021336  012704  000002                 MOV     #2,R4           ;PASS TIME-OUT OF 2 MILLI SECS.
 5011 021342  004767  172352                 JSR     PC,DELAY        ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
 5012 021346  010301                         MOV     R3,R1           ;RESTORE THE CURRENT LINE NUMBER.
 5013                                      ;+
 5014                                      ; Test the state of the DMA_ABORT bit on the line under test.
 5015                                      ;  Report error if DMA_ABORT bit is set.
 5016                                      ;-
```

HARDWARE TEST - DMABRT -

```
5017 021350  005267  162532                  INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4108.
5018 021354  010177  160662                  MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
5019 021360  032777  000001  160664          BIT     #BIT0,@LNCTRA   ;TEST THE STATE OF THE DMA_ABORT BIT.
5020 021366  001002                        BNE     6$              ;GO REPORT ERROR IF BIT IS SET.
5021 021370  000404                        BR      10$             ;BRANCH TO CHECK FOR ANY MORE LINES TO TEST.
5022                                      ;+
5023                                      ; Report error, skip further testing on this line.
5024                                      ;-
5025 021372  010301                4$:      MOV     R3,R1           ;RESTORE THE CURRENT LINE NUMBER.
5026
5027 021374  012702  006076          6$:      MOV     #EM4102,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
5028                                                                 ; "DMA_ABORT BIT BAD ON LINE nn".
5029 021400                        8$:     ERROR                   ;                >>>>> ERROR <<<<<.
     021400  104460                                                                          TRAP    C$ERROR
5030                                      ;+
5031                                      ; Verify all active lines have been tested.
5032                                      ;-
5033 021402  005201                10$:     INC     R1              ;INCREMENT THE LINE NUMBER COUNTER.
5034 021404  005705                         TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
5035 021406  001244                       BNE     2$              ;YES; BRANCH TO TEST THE NEXT LINE.
5036 021410  000402                       BR      60$             ;NO; EXIT THIS TEST.
5037
5038 021412  004767  174364          50$:    JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5039 021416  005067  160642          60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5040
5041 021422                              ENDTST
     021422                                                                      L10025:
     021422  104401                                                                      TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - OAUTOI -
                                ;+**************************************************************************
                                ;*                      - OAUTO BIT INACTIVE TEST -
                                ;*
                                ;*      This test verifies that the DUT's OAUTO function behaves correctly
                                ;*      when inactive, ie OAUTO bit clear.
                                ;*      This test will only execute if staggered loopback mode is selected.
                                ;*      The special staggered loopback connector must be fitted.
                                ;*
                                ;--**************************************************************************

021424                                  BGNTST
021424                                                                  T4::
021424  126727  160606  000002          CMPB    LOPBCK,#2       ;CHECK MODE SELECTED.
021432  001402                        BEQ     .+6             ;DO NOT EXIT IF STAGGERD LOPBCK MODE SELECTED.
021434  000167  000516                JMP     60$             ;EXIT THIS TEST.
        000004                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
021440  012767  000004  160620          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (49)
021446  012767  177777  160610          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
021454  012767  000001  162422          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
021462  012767  011445  162416          MOV     #4901.,ERRNBR   ;SET ERROR NUMBER TO 4901.
021470  012767  006216  162412          MOV     #EM4901,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
021476  012767  012572  162406          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 >>>>> 4901 <<<<<.
                                ;-
021504  004767  171776                  JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
021510  103402                        BCS     .+6             ;DO NOT EXIT IF RESET WAS SUCCESSFUL.
021512  000167  000440                JMP     60$             ;EXIT THIS TEST.
                                ;+
                                ; Set-up the associated TX/RX line number tables.
                                ;-
021516  004767  171330                  JSR     PC,ASLNTL       ;INITIALISE THE ASSOCIATED TX/RX TABLES.
                                ;+
                                ; Set external loopback, disable OAUTO and enable receiver 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.
                                ;-
021522  016705  160506                  MOV     ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
021526  012700  000004                  MOV     #4,R0           ;PASS THE LNCTRL CONTENTS.
021532  004767  175154                JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
021536  012705  000377                  MOV     #MAPLNS,R5      ;PASS BIT MAP OF ALL LINES.
021542  012700  177670                  MOV     #177670,R0      ;PASS THE LPR CONTENTS.
021546  004767  175170                JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
021552  004767  174454                JSR     PC,TXENBL       ;ENABLE TRANSMITTERS ON ALL LINES.
                                ;+
                                ; Set up outer loop for testing active lines in both line groups.
                                ;-
021556  012703  100000                  MOV     #100000,R3      ;SET-UP LOOP CONTROL FLAG.
021562  016705  160446                  MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
021566  046705  160502                  BIC     LGRP2M,R5       ;REMOVE LINES IN GROUP 2.
021572  010567  000352          2$:     MOV     R5,45$          ;SAVE THE CURRENT LINE GROUP.
021576  005067  000344                  CLR     40$             ;CLEAR THE LINE NUMBER COUNTER.
021602  016701  000340          4$:     MOV     40$,R1          ;COPY THE LINE NUMBER.
```

```
 5099 021606  000241                          CLC                     ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
 5100 021610  006005                          ROR     R5              ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
 5101 021612  103054                        BCC     8$                ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
 5102                                    ;+
 5103                                    ; Test the state of the OAUTO bit on the line under test.
 5104                                    ;  Report error if it is found set, and skip further testing of that line.
 5105                                    ;-
 5106 021614  012767  011446  162264          MOV     #4902.,ERRNBR   ;SET THE ERROR NUMBER TO 4902.
 5107 021622  010177  160414                  MOV     R1,@CSRA        ;SELECT THE LINE TO BE TESTED.
 5108 021626  032777  000020  160416          BIT     #BIT4,@LNCTRA   ;TEST THE STATE OF THE OAUTO BIT.
 5109 021634  001404                        BEQ     6$                ;SKIP ERROR REPORT IF OAUTO BIT IS CLEAR.
 5110 021636  012702  006250                  MOV     #EM4902,R2      ;PASS THE ERROR MESSAGE.
 5111                                                                  ; "OAUTO BIT BAD ON LINE nn"
 5112 021642                                ERROR                     ;               >>>>> ERROR #4902 <<<<<.
      021642  104460                                                                      TRAP    C$ERROR
 5113 021644  000437                        BR      8$                ;SKIP FURTHER TESTING OF THIS LINE.
 5114                                    ;+
 5115                                    ; Transmit the XOFF (ASCII DC3) on the associated line.
 5116                                    ;-
 5117 021646  116177  004044  160366  6$:      MOVB    TXRLNB(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
 5118 021654  012777  100023  160362          MOV     #100023,@TXCHA  ;TRANSMIT THE XOFF CHARACTER TO THE LUT.
 5119                                    ;+
 5120                                    ; Wait for transmission to complete.
 5121                                    ;-
 5122 021662  005267  162220                  INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4903.
 5123 021666  012701  170012                  MOV     #170012,R1      ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
 5124 021672  016702  160344                  MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 5125 021676  004767  174674                JSR     PC,WAIBIS         ;WAIT FOR DMA TO COMPLETE.
 5126 021702  103123                        BCC     50$               ;ABORT TEST IF TIMEOUT OCCURRED.
 5127 021704  012704  000005                  MOV     #5,R4           ;PASS TIME-OUT OF 5 MILLI SECS.
 5128 021710  004767  172004                JSR     PC,DELAY          ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
 5129                                    ;+
 5130                                    ; Test the state of the TX_ENABLE bit on the line under test.
 5131                                    ;  Report error if TX_ENABLE bit is clear.
 5132                                    ;-
 5133 021714  005267  162166                  INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4904.
 5134 021720  016701  000222                  MOV     40$,R1          ;GET THE NUMBER OF THE LINE TEST.
 5135 021724  010177  160312                  MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
 5136 021730  005777  160322                  TST     @TXAD2A         ;TEST THE STATE OF THE TX_ENABLE BIT.
 5137 021734  100403                        BMI     8$                ;SKIP ERROR REPORT IF BIT IS SET.
 5138 021736  012702  006250                  MOV     #EM4902,R2      ;PASS THE MESSAGE TO BE REPORTED.
 5139                                                                  ; "OAUTO BIT BAD ON LINE nn".
 5140 021742                                ERROR                     ;               >>>>> ERROR #4904 <<<<<.
      021742  104460                                                                      TRAP    C$ERROR
 5141
 5142 021744  005267  000176          8$:     INC     40$             ;INCREMENT THE LINE NUMBER,
 5143 021750  005705                          TST     R5              ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
 5144 021752  001313                        BNE     4$                ;
 5145                                    ;+
 5146                                    ; Disable transmitters on the selected lines in the current line group.
 5147                                    ;-
 5148 021754  016705  000170                  MOV     45$,R5          ;RESTORE THE CURRENT LINE ACTIVE LINE GROUP.
 5149 021760  004767  174152                JSR     PC,TXDSBL         ;DISABLE TRANSMITTERS ON THE SELECTED LINES.
 5150 021764  016705  000160                  MOV     45$,R5          ;GET THE CURRENT ACTIVE LINE GROUP AGAIN.
 5151 021770  005067  000152                  CLR     40$             ;CLEAR THE LINE COUNTER.
 5152 021774  012767  011451  162104  10$:     MOV     #4905.,ERRNBR   ;SET ERROR NUMBER TO 4905.
 5153 022002  016701  000140                  MOV     40$,R1          ;COPY THE LINE NUMBER.
```

HARDWARE TEST - OAUTOI -

```
5154 022006  000241                    CLC                     ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5155 022010  006005                    ROR     R5              ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5156 022012  103035                    BCC     12$             ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5157                                  ;+
5158                                  ; Transmit the XON (ASCII DC1) on the associated line.
5159                                  ;-
5160 022014  116177  004044  160220    MOVB    TXRLNB(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
5161 022022  012777  100021  160214    MOV     #100021,@TXCHA  ;TRANSMIT THE XON CHARACTER TO THE LUT.
5162                                  ;+
5163                                  ; Wait for transmission to complete.
5164                                  ;-
5165 022030  012701  170012            MOV     #170012,R1      ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
5166 022034  016702  160202            MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5167 022040  004767  174532            JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE.
5168 022044  103042                    BCC     50$             ;ABORT TEST IF TIMEOUT OCCURRED.
5169 022046  012704  000005            MOV     #5,R4           ;PASS TIME-OUT OF 5 MILLI SECS.
5170 022052  004767  171642            JSR     PC,DELAY        ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5171                                  ;+
5172                                  ; Test the state of the TX_ENABLE bit on the line under test.
5173                                  ;  Report error if TX_ENABLE bit is set.
5174                                  ;-
5175 022056  005267  162024            INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4906.
5176 022062  016701  000060            MOV     40$,R1          ;GET THE NUMBER OF THE LINE UNDER TEST.
5177 022066  010177  160150            MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
5178 022072  005777  160160            TST     @TXAD2A         ;TEST THE STATE OF THE TX_ENABLE BIT.
5179 022076  100003                    BPL     12$             ;SKIP ERROR REPORT IF BIT IS CLEAR.
5180 022100  012702  006250            MOV     #EM4902,R2      ;PASS THE MESSAGE TO BE REPORTED.
5181                                                          ; "OAUTO BIT BAD ON LINE nn".
5182 022104                          ERROR                   ;               >>>>> ERROR #4906 <<<<<.
     022104  104460                                                              TRAP    C$ERROR
5183
5184 022106  005267  000034    12$:    INC     40$             ;INCREMENT THE LINE NUMBER,
5185 022112  005705                    TST     R5              ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5186 022114  001327                    BNE     10$             ;
5187                                  ;+
5188                                  ; Check loop control flag to determine if both sets of lines have been tested
5189                                  ;  If this is the fist time around, re-enable TX on all lines, generate active
5190                                  ;  bit map for second line group.
5191                                  ;-
5192 022116  005703                    TST     R3              ;HAVE BOTH LINE GROUPS BEEN TESTED?.
5193 022120  001416                    BEQ     60$             ;YES; THEN EXIT THIS TEST.
5194 022122  005003                    CLR     R3              ;NO; CLEAR THE LOOP CONTROL FLAG,
5195 022124  012705  000377            MOV     #MAPLNS,R5      ;PASS THE BIT MAP OF ALL AVAILABLE LINE.
5196 022130  004767  174076            JSR     PC,TXENBL       ;RE-ENABLE TRANSMISSION ON ALL LINES.
5197 022134  016705  160074            MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
5198 022140  046705  160126            BIC     LGRP1M,R5       ;REMOVE ALL ACTIVE LINES IN GROUP 1.
5199 022144  000612                    BR      2$              ;ONCE MORE AROUND AND WE ARE DONE.
5200
5201 022146  000000            40$:    .WORD   0               ;STORAGE FOR CURRENT LINE NUMBER.
5202 022150  000000            45$:    .WORD   0               ;STORAGE FOR CURRENT ACTIVE LINE BIT MAP.
5203 022152  004767  173624    50$:    JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5204 022156  005067  160102    60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5205
5206 022162                            ENDTST
     022162                                                                  L10026:
     022162  104401                                                              TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - OAUTOA -
                                ;+*****************************************************************************
                                ;*                        - OAUTO BIT ACTIVE TEST -
                                ;*
                                ;*      This test verifies that the DUT's OAUTO function behaves correctly
                                ;*      when active, ie OAUTO bit asserted high.
                                ;*      This test will only execute if the staggered loopback mode is selected.
                                ;*      The special staggered loopback connector must be fitted.
                                ;*
                                ;--*****************************************************************************

022164                                  BGNTST
022164                                                                          T5::
022164  126727  160046  000002          CMPB    LOPBCK,#2       ;CHECK MODE SELECTED.
022172  001402                          BEQ     .+6             ;DO NOT EXIT IF STAGGERD LOPBCK MODE SELECTED.
022174  000167  000516                  JMP     60$             ;EXIT THIS TEST.
000005                                  TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
022200  012767  000005  160060          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (50)
022206  012767  177777  160050          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
022214  012767  000001  161662          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
022222  012767  011611  161656          MOV     #5001.,ERRNBR   ;SET ERROR NUMBER TO 5001.
022230  012767  006302  161652          MOV     #EM5001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
022236  012767  012572  161646          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 >>>>> 5001 <<<<<.
                                ;-
022244  004767  171236                  JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
022250  103402                          BCS     .+6             ;DO NOT EXIT IF RESET WAS SUCCESSFUL.
022252  000167  000440-                 JMP     60$             ;EXIT THIS TEST.
                                ;+
                                ; Set-up the associated TX/RX line number tables.
                                ;-
022256  004767  170570                  JSR     PC,ASLNTL       ;INITIALISE THE ASSOCIATED TX/RX TABLES.
                                ;+
                                ; Set external loopback,enable OAUTO and receiver functions 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.
                                ;-
022262  016705  157746                  MOV     ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
022266  012700  000024                  MOV     #24,R0          ;PASS THE LNCTRL CONTENTS.
022272  004767  174414                  JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
022276  012705  000377                  MOV     #MAPLNS,R5      ;PASS BIT MAP OF ALL LINES.
022302  012700  177670                  MOV     #177670,R0      ;PASS THE LPR CONTENTS.
022306  004767  174430                  JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
022312  004767  173714                  JSR     PC,TXENBL       ;ENABLE TRANSMITTERS ON ALL LINES.
                                ;+
                                ; Set up outer loop for testing active lines in both line groups.
                                ;-
022316  012703  100000                  MOV     #100000,R3      ;SET-UP LOOP CONTROL FLAG.
022322  016705  157706                  MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
022326  046705  157742                  BIC     LGRP2M,R5       ;REMOVE LINES IN GROUP 2.
022332  010567  000352          2$:     MOV     R5,45$          ;SAVE THE CURRENT LINE GROUP.
022336  005067  000344                  CLR     40$             ;CLEAR THE LINE NUMBER COUNTER.
```

```
 5264 022342  016701  000340          4$:     MOV     40$,R1          ;COPY THE LINE NUMBER.
 5265 022346  000241                          CLC                     ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
 5266 022350  006005                          ROR     R5              ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
 5267 022352  103054                        BCC     8$              ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
 5268                                    ;+
 5269                                    ; Test the state of the OAUTO bit on the line under test.
 5270                                    ;  Report error if it is found clear, and skip further testing of that line.
 5271                                    ;-
 5272 022354  012767  011612  161524          MOV     #5002.,ERRNBR   ;SET THE ERROR NUMBER TO 5002.
 5273 022362  010177  157654                  MOV     R1,@CSRA        ;SELECT THE LINE TO BE TESTED.
 5274 022366  032777  000020  157656          BIT     #BIT4,@LNCTRA   ;TEST THE STATE OF THE OAUTO BIT.
 5275 022374  001004                        BNE     6$              ;SKIP ERROR REPORT IF OAUTO BIT IS SET.
 5276 022376  012702  006250                  MOV     #EM4902,R2      ;PASS THE ERROR MESSAGE.
 5277                                                                ;  "OAUTO BIT BAD ON LINE nn"
 5278 022402                                 ERROR                   ;                >>>>> ERROR #5002 <<<<<.
      022402  104460                                                                           TRAP    C$ERROR
 5279 022404  000437                        BR      8$              ;SKIP FURTHER TESTING OF THIS LINE.
 5280                                    ;+
 5281                                    ; Transmit the XOFF (ASCII DC3) on the associated line.
 5282                                    ;-
 5283 022406  116177  004044  157626  6$:     MOVB    TXRLNB(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
 5284 022414  012777  100023  157622          MOV     #100023,@TXCHA  ;TRANSMIT THE XOFF CHARACTER TO THE LUT.
 5285                                    ;+
 5286                                    ; Wait for transmission to complete.
 5287                                    ;-
 5288 022422  005267  161460                  INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 5003.
 5289 022426  012701  170012                  MOV     #170012,R1      ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
 5290 022432  016702  157604                  MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 5291 022436  004767  174134                JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE.
 5292 022442  103123                        BCC     50$             ;ABORT TEST IF TIMEOUT OCCURRED.
 5293 022444  012704  000005                  MOV     #5,R4           ;PASS TIME-OUT OF 5 MILLI SECS.
 5294 022450  004767  171244                JSR     PC,DELAY        ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
 5295                                    ;+
 5296                                    ; Test the state of the TX_ENABLE bit on the line under test.
 5297                                    ;  Report error if TX_ENABLE bit is set.
 5298                                    ;-
 5299 022454  005267  161426                  INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 5004.
 5300 022460  016701  000222                  MOV     40$,R1          ;GET THE NUMBER OF THE LINE TEST.
 5301 022464  010177  157552                  MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
 5302 022470  005777  157562                  TST     @TXAD2A         ;TEST THE STATE OF THE TX_ENABLE BIT.
 5303 022474  100003                        BPL     8$              ;SKIP ERROR REPORT IF BIT IS CLEAR.
 5304 022476  012702  006250                  MOV     #EM4902,R2      ;PASS THE MESSAGE TO BE REPORTED.
 5305                                                                ;  "OAUTO BIT BAD ON LINE nn".
 5306 022502                                 ERROR                   ;                >>>>> ERROR #5004 <<<<<.
      022502  104460                                                                           TRAP    C$ERROR
 5307
 5308 022504  005267  000176          8$:     INC     40$             ;INCREMENT THE LINE NUMBER,
 5309 022510  005705                          TST     R5              ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
 5310 022512  001313                        BNE     4$              ;
 5311                                    ;+
 5312                                    ; Disable transmitters on the selected lines in the current line group.
 5313                                    ;-
 5314 022514  016705  000170                  MOV     45$,R5          ;RESTORE THE CURRENT LINE ACTIVE LINE GROUP.
 5315 022520  004767  173412                JSR     PC,TXDSBL       ;DISABLE TRANSMITTERS ON THE SELECTED LINES.
 5316 022524  016705  000160                  MOV     45$,R5          ;GET THE CURRENT LINE ACTIVE LINE GROUP AGAIN.
 5317 022530  005067  000152                  CLR     40$             ;CLEAR THE LINE COUNTER.
 5318 022534  012767  011615  161344  10$:     MOV     #5005.,ERRNBR   ;SET ERROR NUMBER TO 5005.
```

```
HARDWARE TEST            - OAUTOA -

 5319 022542  016701  000140                 MOV     40$,R1          ;COPY THE LINE NUMBER.
 5320 022546  000241                         CLC                     ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
 5321 022550  006005                         ROR     R5              ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
 5322 022552  103035                       BCC     12$             ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
 5323                                     ;+
 5324                                     ; Transmit the XON (ASCII DC1) on the associated line.
 5325                                     ;-
 5326 022554  116177  004044  157460         MOVB    TXRLNB(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
 5327 022562  012777  100021  157454         MOV     #100021,@TXCHA  ;TRANSMIT THE XON CHARACTER TO THE LUT.
 5328                                     ;+
 5329                                     ; Wait for transmission to complete.
 5330                                     ;-
 5331 022570  012701  170012                 MOV     #170012,R1      ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
 5332 022574  016702  157442                 MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 5333 022600  004767  173772                JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE.
 5334 022604  103042                       BCC     50$             ;ABORT TEST IF TIMEOUT OCCURRED.
 5335 022606  012704  000005                 MOV     #5,R4           ;PASS TIME-OUT OF 5 MILLI SECS.
 5336 022612  004767  171102                JSR     PC,DELAY        ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
 5337                                     ;+
 5338                                     ; Test the state of the TX_ENABLE bit on the line under test.
 5339                                     ;  Report error if TX_ENABLE bit is clear.
 5340                                     ;-
 5341 022616  005267  161264                 INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 5006.
 5342 022622  016701  000060                 MOV     40$,R1          ;GET THE NUMBER OF THE LINE UNDER TEST.
 5343 022626  010177  157410                 MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
 5344 022632  005777  157420                 TST     @TXAD2A         ;TEST THE STATE OF THE TX_ENABLE BIT.
 5345 022636  100403                       BMI     12$             ;SKIP ERROR REPORT IF BIT IS CLEAR.
 5346 022640  012702  006250                 MOV     #EM4902,R2      ;PASS THE MESSAGE TO BE REPORTED.
 5347                                                                 ; "OAUTO BIT BAD ON LINE nn".
 5348 022644                                 ERROR                   ;                >>>>> ERROR #5006 <<<<<.
      022644  104460                                                                             TRAP    C$ERROR
 5349
 5350 022646  005267  000034         12$:    INC     40$             ;INCREMENT THE LINE NUMBER,
 5351 022652  005705                         TST     R5              ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
 5352 022654  001327                       BNE     10$             ;
 5353                                     ;+
 5354                                     ; Check loop control flag to determine if both sets of lines have been tested
 5355                                     ;  If this is the fist time around, re-enable TX on all lines, generate active
 5356                                     ;  bit map for second line group.
 5357                                     ;-
 5358 022656  005703                         TST     R3              ;HAVE BOTH LINE GROUPS BEEN TESTED?.
 5359 022660  001416                       BEQ     60$             ;YES; THEN EXIT THIS TEST.
 5360 022662  005003                         CLR     R3              ;NO; CLEAR THE LOOP CONTROL FLAG,
 5361 022664  012705  000377                 MOV     #MAPLNS,R5      ;PASS THE BIT MAP OF ALL AVAILABLE LINE.
 5362 022670  004767  173336                JSR     PC,TXENBL       ;RE-ENABLE TRANSMISSION ON ALL LINES.
 5363 022674  016705  157334                 MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
 5364 022700  046705  157366                 BIC     LGRP1M,R5       ;REMOVE ALL ACTIVE LINES IN GROUP 1.
 5365 022704  000612                       BR      2$              ;ONCE MORE AROUND AND WE ARE DONE.
 5366
 5367 022706  000000         40$:    .WORD   0               ;STORAGE FOR CURRENT LINE NUMBER.
 5368 022710  000000         45$:    .WORD   0               ;STORAGE FOR CURRENT ACTIVE LINE BIT MAP.
 5369 022712  004767  173064 -       50$:    JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
 5370 022716  005067  157342         60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 5371
 5372 022722                                 ENDTST
      022722                                                                         L10027:
      022722  104401                                                                         TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST          - IAUTOI -
                                ;++*************************************************************************
                                ;*                      - IAUTO BIT INACTIVE TEST -
                                ;*
                                ;*      This test verifies that the DUT's IAUTO function behaves correctly
                                ;*      when inactive, ie. IAUTO bit clear.
                                ;*      All active lines are tested individually by filling the FIFO
                                ;*      then reading the received data checking for the presence of
                                ;*      XOFF(ASCII DC3) or XON (ASCII DC1) characters.
                                ;*      If any are found then appropriate errors are reported.
                                ;*      Any BMP codes that are found will be placed on the BMP code queue,
                                ;*      to be reported later.
                                ;*      The characters are transmitted on all active lines, in internal
                                ;*      loopback mode.
                                ;*
                                ;--*************************************************************************

 022724                                 BGNTST
 022724                                                                 T6::
        000006                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 022724 012767  000006  157334   MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (51)
 022732 012767  177777  157324   MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 022740 012767  000001  161136   MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 022746 012767  011755  161132   MOV    #5101.,ERRNBR   ;SET ERROR NUMBER TO 5101.
 022754 012767  006332  161126   MOV    #EM5101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
 022762 012767  012572  161122   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 >>>>> 5101 <<<<<.
                                ;-
 022770  004767  170512         JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
 022774  103146                 BCC     60$             ;EXIT TEST IF FATAL ERROR FOUND.

                                ;+
                                ; Initialize the 256 byte data pattern.
                                ;  Ensure the data pattern is free from XON's or XOFF's to prevent errors.
                                ;  Note: the first two characters and the last two characters will be the same.
                                ;-
 022776  004767  171242         JSR     PC,INDTPX       ;INITIALISE DATA PATTERN.
                                ;+
                                ; Set internal loopback, disable IAUTO, enable receiver on the selected line.
                                ; Set LPR to 38.4k baud, 8 bits per character, odd parity, 2 stop bits.
                                ;-
 023002  016705  157226          MOV    ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
 023006  012700  000204          MOV    #204,R0         ;PASS INT'L LOPBCK, ENABLE RX, DISABLE IAUTO.
 023012  004767  173674         JSR     PC,WTWLNC       ;INITIALISE THE LINE CONTROL REGISTER.
 023016  012700  177670          MOV    #177670,R0      ;PASS THE LPR CONTENTS.
 023022  004767  173714         JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
 023026  012704  000012          MOV    #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
 023032  004767  170662         JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.

                                ;+
                                ; Set up loop for all active lines.
                                ; Test the state of the IAUTO bit prior to transmitting the data pattern.
                                ;  If the bit is set, then report the error and skip transmitting
```

```
                                        ;  the data pattern on the selected line.
                                        ; Transmit a 256 character data pattern using DMA, on a single channel
                                        ; Empty the fifo, and verify no XOFF or XON chars were found.
                                        ;-
023036  005001                                CLR     R1              ;CLEAR THE LINE NUMBER COUNTER.
023040  005067  000244                      CLR     55$             ;CLEAR STORAGE FOR LINE NUMBER.
023044  012767  011756  161034    2$:       MOV     #5102.,ERRNBR   ;SET THE ERROR NUMBER TO 5102.
023052  004767  171756                    JSR     PC,PUFIFO       ;PURGE THE FIFO.
023056  103111                            BCC     50$             ;GO REPORT ERROR IF FIFO DID NOT PURGE.
023060  000241                             CLC                     ;CLEAR CARRY PRIOR TO ROTATING BIT MAP.
023062  006005                             ROR     R5              ;ROTATE THE BIT MAP INTO THE CARRY BIT.
023064  103077                            BCC     12$             ;BRANCH IF LINE IS INACTIVE.
                                    ;+
                                    ; Test the IAUTO bit on the selected active line.
                                    ; Report error if it is set.
                                    ;  Do not transmit the data pattern on the selected line.
                                    ;-
023066  005267  161014                       INC     ERRNBR          ;SET ERROR NUMBER TO 5103.
023072  010177  157144                       MOV     R1,@CSRA        ;SELECT LINE TO TEST.
023076  032777  000002  157146               BIT     #BIT1,@LNCTRA   ;TEST THE STATE OF THE IAUTO BIT ON THIS LINE.
023104  001404                             BEQ     4$              ;SKIP ERROR IF IAUTO BIT CLEAR.
023106  012702  006360                       MOV     #EM5102,R2      ;PASS THE CORRECT ERROR MESSAGE.
023112                                       ERROR                   ;                >>>>> ERROR <<<<<.
023112  104460                                                                                TRAP    C$ERROR
023114  000463                             BR      12$             ;SKIP TRANSMITTING DATA PATTERN.

                                    ;+
                                    ; Transmit data pattern of 256 chars.
                                    ;-
023116  005267  160764          4$:          INC     ERRNBR          ;SET ERROR NUMBER TO 5104.
023122  012702  002704                       MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
023126  012703  000400                       MOV     #256.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
023132  004767  170622                     JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
023136  103061                              BCC     50$             ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.

                                    ;+
                                    ; Wait for DMA to complete, then wait for the last character plus XOFF
                                    ; to arrive in the fifo.
                                    ;-
023140  005267  160742                       INC     ERRNBR          ;SET ERROR NUMBER TO 5105.
023144  012701  170536                       MOV     #170536,R1      ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
023150  016702  157066                       MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
023154  004767  173416                     JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
023160  103050                             BCC     50$             ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
023162  012704  000012                      MOV     #10.,R4         ;PASS DELAY OF 10 MILLI SECS.
023166  004767  170526                     JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.

                                    ;+
                                    ; Read 256 chars from the fifo.  Report error if any XOFF's or XON's
                                    ; are found.
                                    ;-
023172  005267  160710                       INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 5106.
023176  012701  000400                       MOV     #256.,R1        ;INITIALISE THE READ COUNTER.
023202  017702  157036          6$:          MOV     @RBUFA,R2       ;READ CHAR FROM THE FIFO.
023206  100035                             BPL     50$             ;GO REPORT ERROR IF FIFO EMPTY.
                                      ;+
                                      ; Check for BMP code in the fifo.  Save any found on the queue.
```

```
 5486                                        ;-
 5487 023210  012700  170301                  MOV    #170301,R0      ;SET UP BMP BIT MASK.
 5488 023214  040200                          BIC    R2,R0           ;TRY TO CLEAR ALL THE BMP BITS.
 5489 023216  001002                         BNE    8$              ;SKIP BMPSAV IF NOT A BMP CODE.
 5490 023220  004767  172320                 JSR    PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
 5491                                        ;+
 5492                                        ; Check for XOFF and XON characters.
 5493                                        ;-
 5494 023224  120227  000023          8$:     CMPB   R2,#23          ;IS IT AN XOFF CHARACTER?.
 5495 023230  001406                         BEQ    10$             ;YES; GO REPORT ERROR.
 5496 023232  120227  000021                  CMPB   R2,#21          ;NO; IS IT AN XON CHARACTER?.
 5497 023236  001403                         BEQ    10$             ;YES; GO REPORT ERROR.
 5498 023240  005301                          DEC    R1              ;DECREMENT THE READ COUNT.
 5499 023242  001357                         BNE    6$              ;LOOP TO READ THE NEXT CHAR.
 5500 023244  000407                         BR     12$             ;GO CHECK FOR ANY UNTESTED ACTIVE LINES.
 5501
 5502 023246  005267  160634          10$:    INC    ERRNBR          ;SET ERROR NUMBER TO 5107.
 5503 023252  016701  000032                  MOV    55$,R1          ;PASS THE LINE NUMBER TO BE REPORTED.
 5504 023256  012702  006416                  MOV    #EM5103,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
 5505 023262                                  ERROR                  ;                >>>>> ERROR <<<<<.
      023262  104460                                                                          TRAP    C$ERROR
 5506                                        ;+
 5507                                        ; Check if all active lines have been tested.
 5508                                        ;-
 5509 023264  005267  000020          12$:    INC    55$             ;INCREMENT LINE NUMBER.
 5510 023270  016701  000014                  MOV    55$,R1          ;GET NUMBER OF THE NEXT LINE TO TEST.
 5511 023274  005705                          TST    R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
 5512 023276  001262                         BNE    2$              ;LOOP TO CHECK NEXT LINE.
 5513 023300  000404                         BR     60$             ;EXIT TEST.
 5514
 5515 023302  004767  172474          50$:    JSR    PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
 5516 023306  000401                         BR     60$             ;EXIT THIS TEST.
 5517 023310  000000          55$:    .WORD  0                       ;STORAGE FOR LINE NUMBER.
 5518 023312  005067  156746          60$:    CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 5519
 5520 023316                                 ENDTST
      023316                                                                    L10030:
      023316  104401                                                                    TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - IAUTOA -
                                ;+*****************************************************************************
                                ;*                        - IAUTO BIT ACTIVE TEST -
                                ;*
                                ;*      This test verifies that the DUT's IAUTO function behaves correctly
                                ;*      when active, ie IAUTO asserted high.
                                ;*      All active lines are tested individually by filling the FIFO, and
                                ;*      checking for the presence of at least one XOFF(ASCII DC3) character
                                ;*      and one XON (ASCII DC1) character.
                                ;*      Any BMP codes that are found will be placed on the BMP code queue,
                                ;*      to be reported later.
                                ;*      The characters are transmitted on all active lines, in internal
                                ;*      loopback mode.
                                ;*
                                ;--*****************************************************************************

023320                                  BGNTST
023320                                                                  T7::
        000007                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
023320  012767  000007  156740           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (52)
023326  012767  177777  156730           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
023334  012767  000001  160542           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
023342  012767  012121  160536           MOV    #5201.,ERRNBR   ;SET ERROR NUMBER TO 5201.
023350  012767  006446  160532           MOV    #EM5201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
023356  012767  012572  160526           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 >>>>> 5201 <<<<<.
                                ;-
023364  004767  170116                   JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
023370  103161                           BCC     60$             ;EXIT TEST IF FATAL ERROR FOUND.

                                ;+
                                ; Initialize the 256 byte data pattern.
                                ;  Ensure the data pattern is free from XON's or XOFF's to prevent errors.
                                ;  Note: the first two characters and the last two characters will be the same.
                                ;-
023372  004767  170646                   JSR     PC,INDTPX       ;INITIALISE DATA PATTERN.
                                ;+
                                ; Set internal loopback, enable IAUTO and receiver on the selected line.
                                ; Set LPR to 38.4k baud, 8 bits per character, odd parity, 2 stop bits.
                                ;-
023376  016705  156632                    MOV    ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
023402  012700  000206                    MOV    #206,R0         ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
023406  004767  173300                   JSR     PC,WTWLNC       ;INITIALISE THE LINE CONTROL REGISTER.
023412  012700  177670                    MOV    #177670,R0      ;PASS THE LPR CONTENTS.
023416  004767  173320                   JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
023422  012704  000012                    MOV    #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
023426  004767  170266                   JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.

                                ;+
                                ; Set up loop for all active lines.
                                ; Test the state of the OAUTO bit prior to transmitting the data pattern.
                                ;  If the bit is clear, then report the error and skip transmitting
                                ;  the data pattern on the selected line.
                                ; Transmit a 224 character data pattern using DMA, on a single channel
```

```
                                        ; Empty the fifo, and count the XOFF and an XON chars found.
                                        ;-
023432  005001                            CLR     R1              ;CLEAR THE LINE NUMBER COUNTER.
023434  005067  000272                    CLR     55$             ;CLEAR STORAGE FOR LINE NUMBER.
023440  012767  012122  160440  2$:       MOV     #5202.,ERRNBR   ;SET THE ERROR NUMBER TO 5202.
023446  004767  171362                    JSR     PC,PUFIFO       ;PURGE THE FIFO.
023452  103124                            BCC     50$             ;GO REPORT ERROR IF FIFO DID NOT PURGE.
023454  000241                            CLC                     ;CLEAR CARRY PRIOR TO ROTATING BIT MAP.
023456  006005                            ROR     R5              ;ROTATE THE BIT MAP INTO THE CARRY BIT.
023460  103112                            BCC     16$             ;BRANCH IF LINE IS INACTIVE.
                                    ;+
                                    ; Test the IAUTO bit on the selected active line.
                                    ; Report error if it is clear.
                                    ;  Do not transmit the data pattern on the selected line.
                                    ;-
023462  005267  160420                    INC     ERRNBR          ;SET ERROR NUMBER TO 5203.
023466  010177  156550                    MOV     R1,@CSRA        ;SELECT LINE TO TEST.
023472  032777  000002  156552            BIT     #BIT1,@LNCTRA   ;TEST THE STATE OF THE IAUTO BIT ON THIS LINE.
023500  001004                            BNE     4$              ;SKIP ERROR IF IAUTO BIT SET.
023502  012702  006472                    MOV     #EM5202,R2      ;PASS THE CORRECT ERROR MESSAGE.
                                                                    ; "IAUTO BIT FOUND CLEAR ON LINE nn"
023506                                  ERROR                   ;                >>>>> ERROR <<<<<.
023506  104460                                                                            TRAP    C$ERROR
023510  000476                            BR      16$             ;SKIP TRANSMITTING DATA PATTERN.

                                    ;+
                                    ; Transmit data pattern to fill the fifo, 223 chars + 32 xoff's + xon.
                                    ;-
023512  005267  160370          4$:       INC     ERRNBR          ;SET ERROR NUMBER TO 5204.
023516  012702  002704                    MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
023522  012703  000337                    MOV     #223.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
023526  004767  170226                    JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
023532  103074                            BCC     50$             ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.

                                    ;+
                                    ; Wait for DMA to complete, then wait for the last character plus XOFF
                                    ; to arrive in the fifo.
                                    ;-
023534  005267  160346                    INC     ERRNBR          ;SET ERROR NUMBER TO 5205.
023540  012701  170536                    MOV     #170536,R1      ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
023544  016702  156472                    MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
023550  004767  173022                    JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
023554  103063                            BCC     50$             ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.

                                    ;+
                                    ; Read 256 chars from the fifo, count any XOFF or XON chars found.
                                    ;-
023556  005003                            CLR     R3              ;CLEAR XOFF AND XON INDICATOR.
023560  005267  160322                    INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 5206.
023564  012701  000400                    MOV     #256.,R1        ;INITIALISE THE READ COUNTER.
023570  017702  156450          6$:       MOV     @RBUFA,R2       ;READ CHAR FROM THE FIFO.
023574  100053                            BPL     50$             ;GO REPORT ERROR IF FIFO EMPTY.
                                    ;+
                                    ; Check for BMP code in the fifo.  Save any found on the queue.
                                    ;-
023576  012700  170301                    MOV     #170301,R0      ;SET UP BMP BIT MASK.
023602  040200                            BIC     R2,R0           ;TRY TO CLEAR ALL THE BMP BITS.
```

```
 5634 023604  001002                    BNE     8$              ;SKIP BMPSAV IF NOT A BMP CODE.
 5635 023606  004767  171732            JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
 5636                                   ;+
 5637                                   ; Check for XOFF and XON characters.
 5638                                   ;-
 5639 023612  120227  000023    8$:      CMPB    R2,#23          ;IS IT AN XOFF CHARACTER?.
 5640 023616  001002                    BNE     10$             ;NO, BRANCH TO SEE IF IT IS AN XON.
 5641 023620  052703  000001            BIS     #BIT0,R3        ;INDICATE THE XOFF CHAR.
 5642 023624  120227  000021   10$:      CMPB    R2,#21          ;IS IT AN XON CHARACTER?.
 5643 023630  001002                    BNE     12$             ;NO, SKIP THE NEXT INSTRUCTION.
 5644 023632  052703  000002            BIS     #BIT1,R3        ;INDICATE THE XON CHAR.
 5645 023636  005301            12$:    DEC     R1              ;DECREMENT THE READ COUNT.  250 READ?
 5646 023640  001410                    BEQ     14$             ;YES, EXIT THE LOOP.
 5647 023642  020127  000300            CMP     R1,#192.        ;NO.  HAVE WE READ EXACTLY 192 CHARS?
 5648 023646  001350                    BNE     6$              ;NO, LOOP TO READ MORE.
 5649 023650  012704  000012            MOV     #10.,R4         ;YES.  DELAY 10MS TO WAIT FOR THE XON CHAR
 5650 023654  004767  170040            JSR     PC,DELAY        ;  (WHICH IS LAST CHAR) TO GET INTO FIFO.
 5651 023660  000743                    BR      6$              ;LOOP TO READ MORE CHARS.
 5652                                   ;+
 5653                                   ; Verify than at least 1 XOFF and 1 XON was found in the fifo.
 5654                                   ; Report error if none were found.
 5655                                   ;-
 5656 023662  020327  000003    14$:     CMP     R3,#3           ;DID WE GET AT LEAST 1 XON AND 1 XOFF?
 5657 023666  001407                    BEQ     16$             ;YES, SKIP ERROR REPORT.
 5658 023670  005267  160212            INC     ERRNBR          ;NO, SET ERROR NUMBER TO 5207.
 5659 023674  016701  000032            MOV     55$,R1          ;PASS THE LINE NUMBER TO BE REPORTED.
 5660 023700  012702  006416            MOV     #EM5103,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
 5661                                                           ; "IAUTO BIT BAD ON LINE nn".
 5662 023704                            ERROR                   ;               >>>>> ERROR <<<<<.
      023704  104460                                                                    TRAP    C$ERROR
 5663                                   ;+
 5664                                   ; Check if all active lines have been tested.
 5665                                   ;-
 5666 023706  005267  000020    16$:     INC     55$             ;INCREMENT LINE NUMBER.
 5667 023712  016701  000014            MOV     55$,R1          ;GET NUMBER OF THE NEXT LINE TO TEST.
 5668 023716  005705                    TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
 5669 023720  001247                    BNE     2$              ;LOOP TO CHECK NEXT LINE.
 5670 023722  000404                    BR      60$             ;EXIT TEST.
 5671
 5672 023724  004767  172052    50$:     JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
 5673 023730  000401                    BR      60$             ;EXIT THIS TEST.
 5674 023732  000000    55$:     .WORD   0               ;STORAGE FOR LINE NUMBER.
 5675 023734  005067  156324    60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 5676
 5677 023740                            ENDTST
      023740                                                                    L10031:
      023740  104401                                                                   TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - FIFDAT -
                                ;+*****************************************************************************
                                ;*                        - FIFO VALID DATA TEST -
                                ;*
                                ;*      This test verifies that the DUT is capable of holding 256 valid
                                ;*      characters in its fifo.
                                ;*      The characters are transmitted on the first available active line, in
                                ;*      internal loopback mode.
                                ;*      The data found in the fifo is compared with the expected data, and any
                                ;*      discrepancies are reported.
                                ;*      Any BMP code found will invalidate the test and cause it to be aborted.
                                ;*      However the BMP code will be placed on the BMP code queue, to be
                                ;*      reported later.
                                ;*
                                ;--*****************************************************************************

023742                                  BGNTST
023742                                                                  T8::
                000010                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
023742  012767  000010  156316           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (53)
023750  012767  177777  156306           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
023756  012767  000001  160120           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
023764  012767  012265  160114           MOV    #5301.,ERRNBR   ;SET ERROR NUMBER TO 5301.
023772  012767  006530  160110           MOV    #EM5301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; 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 >>>>> 5301 <<<<<.
                                ;-
024000  004767  167502                  JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
024004  103107                          BCC     60$             ;EXIT TEST IF FATAL ERROR FOUND.
                                ;+
                                ; Find an active line on which to perform the test.
                                ; Initialise 256 byte data pattern.
                                ;-
024006  004767  170122                  JSR     PC,FINACT       ;FIND AN ACTIVE LINE.
024012  103104                          BCC     60$             ;EXIT IF NO ACTIVE LINES FOUND.
024014  004767  170174                  JSR     PC,INDATP       ;INITIALISE THE DATA PATTERN.
                                ;+
                                ; Transmit a 265 character data pattern using DMA, on a single channel
                                ; at 38.4k baud, 8 bits per character, odd parity, 2 stop bits.
                                ;-
                                    ;+
                                    ; Set internal loopback on the selected line.
                                    ; Transmit the data pattern on the first available active line.
                                    ;-
024020  012700  000204                   MOV    #204,R0         ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
024024  004767  172662                  JSR     PC,WTWLNC       ;INITILAISE THE LINE CONTROL REGISTER.
024030  012700  177670                   MOV    #177670,R0      ;PASS THE LPR CONTENTS.
024034  004767  172702                  JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
024040  012704  000012                   MOV    #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
024044  004767  167650                  JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
024050  012702  002704                   MOV    #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
024054  012703  000400                   MOV    #BUFMID-BUFBAS,R3 ;PASS THE LENGTH OF THE DATA PATTERN.
024060  005267  160022                   INC    ERRNBR          ;SET ERROR NUMBER TO 5302.
024064  004767  167670                  JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
024070  103053                           BCC    50$             ;ABORT TEST IF ERROR FOUND DURING DMA TX.
```

```
                                        ;+
                                        ; Wait for DMA to complete, then wait for the last character to arrive in
                                        ; the fifo.
                                        ;-
024072  005267  160010                      INC     ERRNBR          ;SET ERROR NUMBER TO 5303.
024076  010103                              MOV     R1,R3           ;SAVE THE NUMBER OF THE SELECTED ACTIVE LINE.
024100  012701  170536                      MOV     #170536,R1      ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
024104  016702  156132                      MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
024110  004767  172462                    JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
024114  103041                            BCC     50$             ;BRANCH IF FIFO EMPTY, ABORT THE TEST.
024116  012704  000005                     MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
024122  004767  167572                    JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
                                    ;+
                                    ; Read the FIFO checking for data corruption, report any errors found.
                                    ; Abort the test if a BMP code was found in the fifo.
                                    ;-
024126  006303                              ASL     R3              ;MULTIPLY BY 2.
024130  005004                              CLR     R4              ;INITIALISE THE EXPECTED DATA.
024132  016705  156106                      MOV     RBUFA,R5        ;GET THE ADDRESS OF THE RECEIVER BUFFER REG.
024136  012767  012270  157742  2$:         MOV     #5304.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
024144  011502                              MOV     (R5),R2         ;GET THE ACTUAL DATA FROM THE FIFO.
024146  100024                            BPL     50$             ;ABORT THE TEST IF THE FIFO IS EMPTY.
                                  ;+
                                  ; Check if the read character is a BMP code.
                                  ;  If it is a BMP code save it on the queue to be reported later, and
                                  ;  abort the test.
                                  ;-
024150  005267  157732                        INC     ERRNBR          ;SET ERROR NUMBER TO 5305.
024154  004767  167226                      JSR     PC,CHKBMP       ;CHECK IF CHARACTER IS A BMP CODE.
024160  103002                              BCC     4$              ;BRANCH IF NOT A BMP CODE.
024162                                      ERROR                   ;                  >>>>> ERROR 5305 <<<<<.
024162  104460                                                                                TRAP    C$ERROR
024164  000417                              BR      60$             ;ABORT THIS TEST.

024166  005267  157714          4$:         INC     ERRNBR          ;SET ERROR NUMBER TO 5306.
024172  120402                              CMPB    R4,R2           ;COMPARE THE EXPECTED WITH THE ACTUAL DATA.
024174  001406                            BEQ     8$              ;SKIP ERROR REPORT IF DATA IS OK.
024176  012767  012424  157706             MOV     #ER9002,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
024204  012701  006555                      MOV     #EM5302,R1      ;PASS THE MESSAGE TO BE REPORTED.
                                        ;REPORT THE ERROR "FIFO BAD, DATA FIELD CORRUPTED"
024210                          6$:     ERROR                   ;                  >>>>> ERROR 5306 <<<<<.
024210  104460                                                                                TRAP    C$ERROR
024212  105204                  8$:       INCB    R4              ;INCREMENT THE EXPECTED DATA.
024214  001350                            BNE     2$              ;LOOP IF NOT DONE.
024216  000402                            BR      60$             ;EXIT

024220  004767  171556          50$:    JSR     PC,TSABRT       ;ABORT THE TEST, REASON SHOWN BY ERROR NUMBER.
024224  005067  156034          60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.

024230                                  ENDTST
024230                                                                          L10032:
024230  104401                                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - FI3QLI -
                                ;+**********************************************************************
                                ;*                      - FIFO 3/4 LEVEL INACTIVE TEST -
                                ;*
                                ;*      This test verifies that the DUT's fifo 3/4 level alarm system
                                ;*      remains inactive while it contains 191 characters or less.
                                ;*      The test looks for an XOFF (ASCII DC3) character in the fifo.
                                ;*      If any XOFF's are found an error will be reported and the test aborted.
                                ;*      Any BMP code found will invalidate the test and cause it to be aborted.
                                ;*      However the BMP code will be placed on the BMP code queue, to be
                                ;*      reported later.
                                ;*      The characters are transmitted on the first available active line, in
                                ;*      internal loopback mode.
                                ;*
                                ;--**********************************************************************

024232                                  BGNTST
024232                                                                  T9::
            000011                       TNUM == TNUM + 1           ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
024232  012767  000011  156026           MOV     #TNUM,TSTNUM       ;SET UP THE TEST NUMBER.        (54)
024240  012767  177777  156016           MOV     #-1,CTRLCF         ;INDICATE THAT WE ARE IN A TEST.
024246  012767  000001  157630           MOV     #1,ERRTYP          ;SET FATAL ERROR TYPE IN ERROR TABLE.
024254  012767  012431  157624           MOV     #5401.,ERRNBR      ;SET ERROR NUMBER TO 5401.
024262  012767  006705  157620           MOV     #EM5401,ERRMSG     ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
024270  012767  011636  157614           MOV     #ERO503,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 >>>>> 5401 <<<<<.
                                ;-
024276  004767  167204                  JSR     PC,CLNRST          ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
024302  103111                        BCC     60$                ;EXIT TEST IF FATAL ERROR FOUND.
                                ;+
                                ; Find an active line on which to perform the test.
                                ;-
024304  004767  167624                  JSR     PC,FINACT          ;FIND THE NUMBER OF THE FIRST ACTIVE LINE.
024310  103106                        BCC     60$                ;EXIT IF NO LINES ARE AVAILABLE.

                                ;+
                                ; Initialize the 256 byte data pattern.
                                ;  Ensure the data pattern is free from XON's or XOFF's to prevent errors.
                                ;  Note: the first two characters and the last two characters will be the same.
                                ;-
024312  004767  167726                  JSR     PC,INDTPX          ;INITIALISE THE DATA PATTERN.
                                ;+
                                ; Transmit a 191 character data pattern using DMA, on a single channel
                                ; at 38.4k baud, 8 bits per character, odd parity, 2 stop bits.
                                ;-
                                    ;+
                                    ; Set internal loopback, enable IAUTO and RX on the selected line.
                                    ; Transmit the data pattern on the first available active line.
                                    ;-
024316  012700  000206                   MOV     #206,R0            ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
024322  004767  172364                  JSR     PC,WTWLNC          ;INITILAISE THE LINE CONTROL REGISTER.
024326  012700  177670                   MOV     #177670,R0         ;PASS THE LPR CONTENTS.
024332  004767  172404                  JSR     PC,WTWLPR          ;SET THE LPR CONTENTS TO 38.4K BAUD.
024336  012704  000012                   MOV     #10.,R4            ;PASS DELAY TIME OF 10 MILLI SECONDS.
```

```
 5840 024342  004767  167352                JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
 5841 024346  012702  002704                 MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
 5842 024352  012703  000277                 MOV     #191.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
 5843 024356  004767  167376                JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
 5844 024362  103057                         BCC     50$             ;IF ERROR FOUND DURING DMA THEN ABORT TEST.
 5845
 5846                                     ;+
 5847                                     ; Wait for DMA to complete, then wait for the last character to arrive in
 5848                                     ; the fifo.
 5849                                     ;-
 5850 024364  005267  157516                  INC     ERRNBR          ;SET ERROR NUMBER TO 5402.
 5851 024370  012701  170454                  MOV     #170454,R1      ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
 5852 024374  016702  155642                  MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
 5853 024400  004767  172172                 JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
 5854 024404  103046                         BCC     50$             ;IF FIFO EMPTY, REPORT ERROR, ABORT THE TEST.
 5855 024406  012704  000005                  MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
 5856 024412  004767  167302                 JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
 5857                                     ;+
 5858                                     ; Read the contents of the fifo.  If any of the following conditions occur
 5859                                     ; report the error and abort the test;
 5860                                     ;  Fifo empty too soon.
 5861                                     ;  BMP code found.
 5862                                     ;  Xoff code found.
 5863                                     ;  Extra (192) character found in fifo.
 5864                                     ;-
 5865 024416  005004                          CLR     R4              ;CLEAR THE CHARACTER COUNT.
 5866 024420  016705  155620                  MOV     RBUFA,R5        ;GET THE ADDRESS OF THE RECEIVER BUFFER REG.
 5867 024424  012767  012267  157454  2$:     MOV     #5303.,ERRNBR   ;SET ERROR NUMBER TO 5403.
 5868 024432  011502                          MOV     (R5),R2         ;GET THE ACTUAL DATA FROM THE FIFO.
 5869 024434  100032                         BPL     50$             ;FIFO EMPTY, ABORT TEST.
 5870 024436  005204                          INC     R4              ;COUNT THE CHARACTER.
 5871                                     ;+
 5872                                     ; Check if the read character is a BMP code.
 5873                                     ;  If it is a BMP code save it on the queue to be reported later, and
 5874                                     ;  abort the test.
 5875                                     ;-
 5876 024440  005267  157442                  INC     ERRNBR          ;SET ERROR NUMBER TO 5404.
 5877 024444  004767  166736                 JSR     PC,CHKBMP       ;CHECK IF CHARACTER IS A BMP CODE.
 5878 024450  103001                         BCC     4$              ;BRANCH IF NOT A BMP CODE.
 5879                                         ;REPORT ERROR "BMP CODE FOUND IN FIFO, TEST INVALIDATED".
 5880 024452  000421                         BR      8$              ;REPORT THE ERROR AND ABORT THE TEST.
 5881                                     ;+
 5882                                     ; Check if the character is an XOFF.  Report the error if one is found.
 5883                                     ;-
 5884
 5885 024454  005267  157426          4$:     INC     ERRNBR          ;SET ERROR NUMBER TO 5405.
 5886 024460  122702  000023                  CMPB    #23,R2          ;CHECK IF THE READ DATA IS AN XOFF.
 5887 024464  001003                         BNE     6$              ;BRANCH IF NOT AN XOFF.
 5888 024466  012701  006744                  MOV     #EM5402,R1      ;PASS THE MESSAGE TO BE REPORTED.
 5889                                         ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
 5890 024472  000411                         BR      8$              ;GO REPORT THE ERROR AND ABORT THE TEST.
 5891
 5892 024474  005267  157406          6$:     INC     ERRNBR          ;SET ERROR NUMBER TO 5406.
 5893 024500  020427  000277                  CMP     R4,#191.        ;CHECK IF WE HAVE READ ALL THE CHARACTERS.
 5894 024504  001347                         BNE     2$              ;LOOP BACK TO GET THE NEXT CHARACTER.
 5895 024506  011502                          MOV     (R5),R2         ;TRY TO READ AN EXTRA CHARACTER FROM THE FIFO.
 5896 024510  100006                         BPL     60$             ;EXIT IF NON FOUND.
```

```
 5897 024512  012701  006744                    MOV     #EM5402,R1      ;PASS THE MESSAGE TO BE REPORTED.
 5898                                           ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
 5899
 5900 024516                          8$:       ERROR                   ;       >>>>> ERRORS 5304 THRU 5306 <<<<<.
      024516  104460                                                                      TRAP    C$ERROR
 5901 024520  000402                           BR      60$             ;EXIT THE TEST.
 5902
 5903                                                                   ;       >>>>> ERRORS 5402 AND 5403 <<<<<.
 5904 024522  004767  171254          50$:     JSR     PC,TSABRT       ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
 5905 024526  005067  155532          60$:      CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 5906
 5907 024532                                    ENDTST
      024532                                                                          L10033:
      024532  104401                                                                      TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - FI3QLA -
                                ;+*****************************************************************************
                                ;*                      - FIFO 3/4 LEVEL ACTIVE TEST -
                                ;*
                                ;*      This test verifies that the DUT's fifo 3/4 level alarm system
                                ;*      becomes active when the fifo contains > 192 characters.
                                ;*      The test compares the actual number of XOFF (ASCII DC3)
                                ;*      characters that are found in the fifo with the expected number.
                                ;*      An error will be reported, if the counts are found to differ.
                                ;*      Any BMP code found will invalidate the test and cause it to be aborted.
                                ;*      However the BMP code will be placed on the BMP code queue, to be
                                ;*      reported later.
                                ;*      The characters are transmitted on the first available active line, in
                                ;*      internal loopback mode.
                                ;*
                                ;--*****************************************************************************

024534                                  BGNTST
024534                                                                  T10::
        000012                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
024534  012767  000012  155524   MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (55)
024542  012767  177777  155514   MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
024550  012767  000001  157326   MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
024556  012767  012575  157322   MOV    #5501.,ERRNBR   ;SET ERROR NUMBER TO 5501.
024564  012767  007005  157316   MOV    #EM5501,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; 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 >>>>> 5501 <<<<<.
                                ;-
024572  004767  166710                  JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
024576  103402                        BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
024600  000167  000414                JMP     60$             ;EXIT TEST FATAL ERROR FOUND.
                                ;+
                                ; Find an active line on which to perform the test.
                                ;-
024604  004767  167324                  JSR     PC,FINACT       ;FIND AN ACTIVE LINE.
024610  103402                        BCS     .+6             ;SKIP EXIT OF TEST IF ACTIVE LINE FOUND.
024612  000167  000402                JMP     60$             ;EXIT TEST.
                                ;+
                                ; Initialize the 256 byte data pattern.
                                ;  Ensure the data pattern is free from XON's or XOFF's to prevent errors.
                                ;  Note: the first two characters and the last two characters will be the same.
                                ;-
024616  004767  167422                  JSR     PC,INDTPX       ;INITIALISE DATA PATTERN.
                                ;+
                                ; Transmit a 256 character data pattern using DMA, on a single channel
                                ; at 38.4k baud, 8 bits per character, odd parity, 2 stop bits.
                                ;-
                                    ;+
                                    ; Set internal loopback, enable IAUTO and receiver on the selected line.
                                    ; Transmit the first 191 characters on the first available active line.
                                    ;-
024622  005267  157260          2$:     INC     ERRNBR          ;SET ERROR NUMBER TO 5502.
024626  012700  000206                  MOV     #206,R0         ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
024632  004767  172054                  JSR     PC,WTWLNC       ;INITIALISE THE LINE CONTROL REGISTER.
024636  012700  177670                  MOV     #177670,R0      ;PASS THE LPR CONTENTS.
```

```
 5965 024642  004767  172074                JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
 5966 024646  012704  000012                 MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
 5967 024652  004767  167042                JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
 5968 024656  010105                         MOV     R1,R5           ;COPY THE LINE NUMBER.
 5969 024660  012702  002704                 MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
 5970 024664  012703  000277                 MOV     #191.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
 5971 024670  004767  167064                JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
 5972 024674  103147                         BCC     50$             ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
 5973
 5974                                      ;+
 5975                                      ; Wait for DMA to complete, then wait for the last character to arrive in
 5976                                      ; the fifo.
 5977                                      ;-
 5978 024676  005267  157204                 INC     ERRNBR          ;SET ERROR NUMBER TO 5503.
 5979 024702  012701  170454                 MOV     #170454,R1      ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
 5980 024706  016702  155330                 MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
 5981 024712  004767  171660                JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
 5982 024716  103136                        BCC     50$             ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
 5983 024720  012704  000005                 MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
 5984 024724  004767  166770                JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
 5985                                      ;+
 5986                                      ;  Transmit a null character which will cause an XOFF to be generated.
 5987                                      ;-
 5988 024730  005267  157152                 INC     ERRNBR          ;SET ERROR NUMBER TO 5504.
 5989 024734  010501                         MOV     R5,R1           ;PASS THE LINE NUMBER.
 5990 024736  012702  002704                 MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
 5991 024742  012703  000001                 MOV     #1,R3           ;PASS THE NUMBER OF
 5992 024746  004767  167006                JSR     PC,DODMA        ;TX A NULL CHARACTER TO CAUSE AN XOFF.
 5993 024752  103120                         BCC     50$             ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
 5994
 5995                                      ;+
 5996                                      ; Wait for the XOFF to be received before TX the next 42 characters
 5997                                      ; which will cause a further 21 XOFF's to be generated.
 5998                                      ;-
 5999 024754  005267  157126                 INC     ERRNBR          ;SET ERROR NUMBER TO 5505.
 6000 024760  012701  170012                 MOV     #170012,R1      ;PASS TIME-OUT VALUE OF 10 MILLI SECS.
 6001 024764  016702  155252                 MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
 6002 024770  004767  171602                JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
 6003 024774  103107                        BCC     50$             ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
 6004 024776  012704  000005                 MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
 6005 025002  004767  166712                JSR     PC,DELAY        ;WAIT FOR XOFF TO GET INTO THE FIFO.
 6006                                      ;+
 6007                                      ; Initialise the 256 byte data pattern to all nulls.
 6008                                      ;-
 6009 025006  012702  002704                 MOV     #BUFBAS,R2      ;INITIALIZE THE DATA PATTERN TO BE
 6010 025012  105022                  4$:    CLRB    (R2)+           ;  ALL NULLS.
 6011 025014  020227  003304                 CMP     R2,#BUFMID      ;
 6012 025020  103774                        BLO     4$              ;
 6013
 6014                                      ;+
 6015                                      ; Transmit a further 31 null characters which will cause 31 XOFF's to be
 6016                                      ; generated.
 6017                                      ;-
 6018 025022  005267  157060                 INC     ERRNBR          ;SET ERROR NUMBER TO 5506.
 6019 025026  010501                         MOV     R5,R1           ;PASS THE LINE NUMBER.
 6020 025030  012702  002704                 MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
 6021 025034  012703  000037                 MOV     #31.,R3         ;PASS THE LENGTH OF THE DATA PATTERN.
```

```
 6022 025040  004767  166714                    JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
 6023 025044  103063                             BCC     50$             ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
 6024                                          ;+
 6025                                          ; Wait for the XOFF's and the null characters  to be received.
 6026                                          ;-
 6027 025046  005267  157034                     INC     ERRNBR          ;SET ERROR NUMBER TO 5507.
 6028 025052  012701  170454                     MOV     #170454,R1      ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
 6029 025056  016702  155160                     MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
 6030 025062  004767  171510                    JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
 6031 025066  103052                            BCC     50$             ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
 6032 025070  012704  000005                     MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
 6033 025074  004767  166620                    JSR     PC,DELAY        ;WAIT FOR XOFF TO GET INTO THE FIFO.
 6034                                          ;+
 6035                                          ; Read the fifo until empty, counting the number of XOFF characters
 6036                                          ; that are found.
 6037                                          ;-
 6038 025100  005004                             CLR     R4              ;CLEAR CHARACTER COUNTER.
 6039 025102  005003                             CLR     R3              ;CLEAR THE XOFF FOUND COUNTER.
 6040 025104  012701  170001                     MOV     #170001,R1      ;INDICATE TO TEST DATA.VALID BIT, TIME-OUT 1MS.
 6041 025110  012767  012604  156770  6$:       MOV     #5508.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND THE LOOP.
 6042 025116  016702  155122                     MOV     RBUFA,R2        ;INDICATE TO CHECK RECEIVE BUFFER REGISTER.
 6043 025122  004767  171450                    JSR     PC,WAIBIS       ;WAIT FOR RECEIVED CHAR OR TIME-OUT.
 6044 025126  103032                            BCC     50$             ;GO REPORT ERROR IF FIFO EMPTY.
 6045 025130  005204                             INC     R4              ;COUNT THE CHARACTER.
 6046                                        ;+
 6047                                        ; Check if for BMP codes in the fifo, abort the test if any are found.
 6048                                        ;  Save the BMP code on the queue to be reported later.
 6049                                        ;-
 6050 025132  005267  156750                       INC     ERRNBR          ;SET ERROR NUMBER TO 5509.
 6051 025136  004767  166244                      JSR     PC,CHKBMP       ;CHECK IF WE HAVE GOT A BMP CODE.
 6052 025142  103422                              BCS     12$             ;GO REPORT THE ERROR IF WE FOUND A BMP CODE.
 6053                                        ;+
 6054                                        ; Check for XOFF character.
 6055                                        ;-
 6056 025144  122702  000023             8$:       CMPB    #23,R2          ;CHECK IF THE RECEIVED CHARACTER WAS AN XOFF.
 6057 025150  001001                              BNE     10$             ;BRANCH IF CHARACTER WAS NOT AN XOFF.
 6058 025152  005203                               INC     R3              ;INCREMENT XOFF FOUND COUNT.
 6059                                        ;+
 6060                                        ; Check if all the characters including the XON have been removed.
 6061                                        ;-
 6062 025154  020427  000400             10$:      CMP     R4,#256.        ;CHECK IF WE HAVE REMOVED ALL THE CHARACTERS.
 6063 025160  002753                              BLT     6$              ;GO GET THE NEXT CHAR IF WE HAVE NOT FINISHED.
 6064
 6065                                        ;+
 6066                                        ; Check if the correct number of XOFF's were found in the fifo,
 6067                                        ; report error if count is incorrect.
 6068                                        ;-
 6069 025162  016767  165420  156716               MOV     5510.,ERRNBR    ;SET UP THE ERROR NUMBER TO 5510.
 6070 025170  022703  000040                       CMP     #32.,R3         ;COMPARE EXPECTED XOFF COUNT WITH ACTUAL COUNT.
 6071 025174  001411                              BEQ     60$             ;EXIT TEST IF SUCCESS.
 6072 025176  012767  011636  156706               MOV     #ER0503,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
 6073 025204  012701  006744                       MOV     #EM5402,R1      ;PASS THE MESSAGE TO BE REPORTED.
 6074                                              ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
 6075 025210                             12$:      ERROR                   ;                >>>>> ERROR <<<<<.
      025210  104460                                                                                TRAP    C$ERROR
 6076 025212  000402                              BR      60$             ;ABORT THE TEST.
 6077
```

HARDWARE TEST - FI3QLA -

```
 6078 025214  004767  170562          50$:    JSR     PC,TSABRT       ;REPORT TEST ABORTED.  ERROR # SHOWS REASON.
 6079 025220  005067  155040          60$:     CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 6080
 6081 025224                                  ENDTST
      025224                                                                  L10034:
      025224  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - FI3QAI -
                                ;+*********************************************************************
                                ;*                  - FIFO 3/4 ALARM LEVEL ACTIVE/INACTIVE TEST -
                                ;*
                                ;*      This test verifies that the DUT's fifo 3/4 level alarm system
                                ;*      becomes active and inactive at the correct levels.
                                ;*      Any BMP code found will invalidate the test and cause it to be aborted.
                                ;*      However the BMP code will be placed on the BMP code queue, to be
                                ;*      reported later.
                                ;*      The characters are transmitted on the first available active line, in
                                ;*      internal loopback mode.
                                ;*
                                ;--*********************************************************************

                                        BGNTST
                                                                                        T11::
                 000013                  TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025226  012767  000013  155032           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (56)
025234  012767  177777  155022           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
025242  012767  000001  156634           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
025250  012767  012741  156630           MOV     #5601.,ERRNBR  ;SET ERROR NUMBER TO 5601.
025256  012767  007042  156624           MOV     #EM5601,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; 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 >>>>> 5601 <<<<<.
                                ;-
025264  004767  166216                   JSR     PC,CLNRST      ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
025270  103402                          BCS     2$             ;SKIP EXITING TEST A SUCCESSFUL RESET.
025272  000167  000412                   JMP     60$            ;EXIT THIS TEST.
025276                          2$:
                                ;+
                                ; Find an active line on which to perform the test.
                                ;-
025276  004767  166632                   JSR     PC,FINACT      ;FIND AN ACTIVE LINE.
025302  103402                          BCS     .+6            ;SKIP EXIT OF TEST IF ACTIVE LINE FOUND.
025304  000167  000400                   JMP     60$            ;EXIT TEST.
                                ;+
                                ; Initialize the 256 byte data pattern.
                                ;  Ensure the data pattern is free from XON's or XOFF's to prevent errors.
                                ;  Note: the first two characters and the last two characters will be the same.
                                ;-
025310  004767  166730                   JSR     PC,INDTPX      ;INITIALISE THE DATA PATTERN.
                                ;+
                                ; Transmit a 256 character data pattern using DMA, on a single channel
                                ; at 38.4k baud, 8 bits per character, odd parity, 2 stop bits.
                                ;-
                                    ;+
                                    ; Set internal loopback, enable IAUTO and receiver on the selected line.
                                    ; Transmit the first 191 characters on the first available active line.
                                    ;-
025314  005267  156566                   INC     ERRNBR         ;SET ERROR NUMBER TO 5602.
025320  012700  000206                   MOV     #206,R0        ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
025324  004767  171362                  JSR     PC,WTWLNC       ;INITILAISE THE LINE CONTROL REGISTER.
025330  012700  177670                   MOV     #177670,R0     ;PASS THE LPR CONTENTS.
025334  004767  171402                  JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
025340  012704  000012                   MOV     #10.,R4        ;PASS DELAY TIME OF 10 MILLI SECONDS.
```

```
6139 025344  004767  166350          JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6140 025350  010105                   MOV     R1,R5           ;COPY THE LINE NUMBER.
6141 025352  012702  002704           MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
6142 025356  012703  000277           MOV     #191.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
6143 025362  004767  166372          JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
6144 025366  103146                   BCC     50$             ;EXIT IF ERROR FOUND DURING DMA TX.
6145                                 ;+
6146                                 ; Wait for DMA to complete, then wait for the last character to arrive in
6147                                 ; the fifo.
6148                                 ;-
6149 025370  005267  156512           INC     ERRNBR          ;SET ERROR NUMBER TO 5603.
6150 025374  012701  170454           MOV     #170454,R1      ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6151 025400  016702  154636           MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
6152 025404  004767  171166          JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6153 025410  103135                  BCC     50$             ;BRANCH IF FIFO EMPTY, ABORT THE TEST.
6154 025412  012704  000005           MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
6155 025416  004767  166276          JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6156
6157                                 ;+
6158                                 ;  Transmit a null character which will cause an XOFF to be generated.
6159                                 ;-
6160 025422  005267  156460           INC     ERRNBR          ;SET ERROR NUMBER TO 5604.
6161 025426  010501                   MOV     R5,R1           ;PASS THE LINE NUMBER.
6162 025430  012702  002704           MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
6163 025434  012703  000001           MOV     #1,R3           ;PASS THE NUMBER OF
6164 025440  004767  166314          JSR     PC,DODMA        ;TX A NULL CHARACTER TO CAUSE AN XOFF.
6165 025444  103117                   BCC     50$             ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6166                                 ;+
6167                                 ; Wait for the XOFF to be received before continuing the test.
6168                                 ;-
6169 025446  005267  156434           INC     ERRNBR          ;SET ERROR NUMBER TO 5605.
6170 025452  012701  170012           MOV     #170012,R1      ;PASS TIME-OUT VALUE OF 10 MILLI SECS.
6171 025456  016702  154560           MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
6172 025462  004767  171110          JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6173 025466  103106                  BCC     50$             ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6174 025470  012704  000005           MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
6175 025474  004767  166220          JSR     PC,DELAY        ;WAIT FOR XOFF TO GET INTO THE FIFO.
6176
6177 025500  010577  154536           MOV     R5,@CSRA        ;SELECT THE LINE READY FOR TRANSMISSION.
6178                                 ;+
6179                                 ; Read three characters, transmit one character until the first 192 characters
6180                                 ; have been read from the fifo, ie until the half level is reached.
6181                                 ;  Then read the fifo until empty.
6182                                 ; Count all XOFF's that are detected.
6183                                 ;-
6184 025504  005005                   CLR     R5              ;CLEAR THE TX FLAG.
6185 025506  005004                   CLR     R4              ;CLEAR THE CHARACTER COUNTER.
6186 025510  012703  00030'           MOV     #192.,R3        ;SET UP READ COUNTER FOR THE FIRST 192 CHARS.
6187
6188 025514  012700  000003     4$:   MOV     #3,R0           ;SET READ COUNTER.
6189 025520  012701  170005     6$:   MOV     #170005,R1      ;INDICATE TO TEST DATA.VALID BIT, TIME-OUT 5MS.
6190 025524  016702  154514           MOV     RBUFA,R2        ;INDICATE TO CHECK RECEIVE BUFFER REGISTER.
6191 025530  004767  171042          JSR     PC,WAIBIS       ;WAIT FOR RECEIVED CHAR OR TIME-OUT.
6192 025534  103046                  BCC     14$             ;EXIT LOOP IF TIME-OUT, FIFO EMPTY.
6193 025536  005300                   DEC     R0              ;DECREMENT READ COUNTER.
6194 025540  005303                   DEC     R3              ;DECREMENT CHAR COUNTER.
6195 025542  003002                   BGT     8$              ;SKIP DISBL'G TX IF FIRST 192 CHARS NOT READ.
```

```
 6196 025544  052705  100000                BIS    #BIT15,R5       ;DISABLE ANY FURTHER TRANSMISSIONS.
 6197                                     ;+
 6198                                     ; Check if the read character is a BMP code.
 6199                                     ;  If it is a BMP code save it on the queue to be reported later, and
 6200                                     ;  abort the test.
 6201                                     ;-
 6202 025550  012767  012746  156330  8$:    MOV    #5606.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
 6203 025556  004767  165624               JSR    PC,CHKBMP       ;CHECK IF CHARACTER IS A BMP CODE.
 6204 025562  103446                       BCS    16$             ;GO REPORT ERROR AND ABORT TEST IF BMP FOUND.
 6205                                     ;+
 6206                                     ; Check for XOFF character.  If one is found, count it.
 6207                                     ; Transmit a null character until the first 192 chars have been read.
 6208                                     ;-
 6209 025564  122702  000023          10$:    CMPB   #23,R2          ;CHECK IF THE RECEIVED CHARACTER WAS AN XOFF.
 6210 025570  001001                       BNE    12$             ;BRANCH IF CHARACTER WAS NOT AN XOFF.
 6211 025572  005204                        INC    R4              ;INCREMENT THE XOFF CHAR FOUND COUNTER.
 6212
 6213 025574  005700                  12$:    TST    R0              ;CHECK READ COUNT, TO SEE IF A CHAR CAN BE TX.
 6214 025576  001350                       BNE    6$              ;BRANCH IF 3 CHARS NAVE NOT YET BEEN READ.
 6215 025600  005705                        TST    R5              ;CHECK THE TRANSMISSION ENABLED FLAG.
 6216 025602  100744                       BMI    4$              ;SKIP TRANSMITTING A CHARACTER IF TX DISABLED.
 6217 025604  012777  100000  154432        MOV    #100000,@TXCHA  ;TX A NULL CHARACTER.
 6218 025612  010446                        MOV    R4,-(SP)        ;SAVE THE XOFF COUNT ON THE STACK.
 6219                                   ;+
 6220                                   ; Wait for the character to be received before continuing the test.
 6221                                   ;-
 6222 025614  005267  156266                INC    ERRNBR          ;SET ERROR NUMBER TO 5607.
 6223 025620  012701  170012                MOV    #170012,R1      ;PASS TIME-OUT VALUE OF 10 MILLI SECS.
 6224 025624  016702  154412                MOV    CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
 6225 025630  004767  170742               JSR    PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
 6226 025634  103023                       BCC    50$             ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
 6227 025636  012704  000005                MOV    #5,R4           ;PASS DELAY OF 5 MILLI SECS.
 6228 025642  004767  166052               JSR    PC,DELAY        ;WAIT FOR XOFF TO GET INTO THE FIFO.
 6229 025646  012604                        MOV    (SP)+,R4        ;RESTORE THE XOFF COUNT.
 6230 025650  000721                       BR     4$              ;GO RESET THE READ COUNT AND GET NEXT CHAR.
 6231
 6232                                   ;+
 6233                                   ; Check if the correct number of XOFF's were found in the fifo
 6234                                   ; Report error if count is incorrect.
 6235                                   ;-
 6236 025652  012767  012750  156226  14$:    MOV    #5608.,ERRNBR   ;SET ERROR NUMBER TO 5608.
 6237 025660  020427  000077                CMP    R4,#63.         ;COMPARE THE EXPECTED AND ACTUAL XOFF COUNTS.
 6238 025664  001411                       BEQ    60$             ;EXIT TEST IF SUCCESS.
 6239 025666  012767  011636  156216        MOV    #ER0503,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
 6240 025674  012701  006744                MOV    #EM5402,R1      ;PASS THE MESSAGE TO BE REPORTED.
 6241                                       ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
 6242 025700                          16$:    ERROR                  ;                >>>>> ERROR <<<<<.
      025700  104460                                                                          TRAP    C$ERROR
 6243 025702  000402                       BR     60$             ;EXIT THIS TEST.
 6244
 6245 025704  004767  170072          50$:    JSR    PC,TSABRT       ;REPORT TEST ABORTED.  ERROR # INDICATES FAULT.
 6246 025710  005067  154350          60$:    CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 6247
 6248 025714                               ENDTST
      025714                                                                          L10035:
      025714  104401                                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - FIHAVL -
                                ;+**********************************************************************
                                ;*                  - FIFO HALF LEVEL ACTIVE/INACTIVE TEST -
                                ;*
                                ;*      This test checks that the DUT's fifo half level alarm system
                                ;*      becomes active and inactive at the correct levels.
                                ;*      Any BMP code found will invalidate the test and cause it to be aborted.
                                ;*      However the BMP code will be placed on the BMP code queue, to be
                                ;*      reported later.
                                ;*      The characters are transmitted on the first available active line, in
                                ;*      internal loopback mode.
                                ;*
                                ;--**********************************************************************

 025716                                 BGNTST
 025716                                                                  T12::
        000014                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 025716 012767  000014  154342           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (57)
 025724 012767  177777  154332           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 025732 012767  000001  156144           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 025740 012767  013105  156140           MOV    #5701.,ERRNBR   ;SET ERROR NUMBER TO 5701.
 025746 012767  007110  156134           MOV    #EM5701,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
 025754 012767  011636  156130           MOV    #ER0503,ERRBLK  ;SELECT THE 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 >>>>> 5701 <<<<<.
                                ;-
 025762  004767  165520                 JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
 025766  103402                        BCS     2$              ;SKIP EXITING TEST A SUCCESSFUL RESET.
 025770  000167  000360                JMP     60$             ;EXIT THIS TEST.
 025774                         2$:
                                ;+
                                ; Find an active line on which to perform the test.
                                ;-
 025774  004767  166134                 JSR     PC,FINACT       ;FIND AN ACTIVE LINE.
 026000  103165                        BCC     60$             ;EXIT IF NO ACTIVE LINES AVAILABLE.
                                ;+
                                ; Initialize the 256 byte data pattern.
                                ;  Ensure the data pattern is free from XON's or XOFF's to prevent errors.
                                ;  Note: the first two characters and the last two characters will be the same.
                                ;-
 026002  004767  166236                 JSR     PC,INDTPX       ;INITIALISE THE DATA PATTERN.
                                ;+
                                ; Fill the fifo by transmitting 225 chars (ie 225 + 31 XOFF's).
                                ; Transmit data pattern using DMA, on a single channel
                                ; at 38.4k baud, 8 bits per character, odd parity, 2 stop bits.
                                ;-
                                    ;+
                                    ; Set internal loopback, enable IAUTO and receiver on the selected line.
                                    ; Transmit the 225 characters on the first available active line.
                                    ;-
 026006  005267  156074                  INC     ERRNBR          ;SET ERROR NUMBER TO 5702.
 026012  004767  167640                 JSR     PC,SETPAR       ;SET UP PARAMETERS FOR TRANSMISSION.
 026016  012700  000341                 MOV     #225.,R0        ;PASS LENGTH OF DATA PATTERN.
 026022  004767  170066                 JSR     PC,TXDATP       ;TRANSMIT DATA PATTERN.
 026026  103150                        BCC     50$             ;EXIT IF ERROR FOUND DURING TX.
```

```
                                        MOV     R1,R5           ;COPY THE LINE NUMBER.
                                    ;+
                                    ; Wait for DMA to complete, then wait for the last character to arrive in
                                    ; the fifo.
                                    ;-
                                        INC     ERRNBR          ;SET ERROR NUMBER TO 5703.
                                       JSR     PC,WAITTX        ;WAIT FOR TRANSMISSION TO COMPLETE.
                                       BCC     50$              ;GO REPORT ERROR IF TX FAILED TO COMPLETE.
                                ;+
                                ; Read the first 130 characters from the fifo, if any XON's are found
                                ; report the error. If any BMP codes are found then save them on the queue
                                ; and abort the test.
                                ;-
                                        INC     ERRNBR          ;SET ERROR NUMBER TO 5704.
                                        MOV     #130.,R0        ;PASS THE NUMBER OF CHARS TO READ.
                                       JSR     PC,READBX        ;READ THE FIRST 130 CHARS FROM THE FIFO.
                                       BCC     50$              ;GO REPORT ERROR IF BMP CODE FOUND.
                                        INC     ERRNBR          ;SET ERROR NUMBER TO 5705.
                                        TST     R1              ;CHECK IF AN XON WAS FOUND.
                                       BNE     40$              ;GO REPORT ERROR IF AN XON WAS FOUND.

                                    ;+
                                    ;  Transmit a null character.
                                    ;-
                                        MOV     R5,@CSRA        ;SELECT THE LINE READY FOR TRANSMISSION.
                                        MOV     #100000,@TXCHA  ;TRANSMIT A NULL CHARACTER.
                                        INC     ERRNBR          ;SET ERROR NUMBER TO 5706.
                                       JSR     PC,WAITTX        ;WAIT FOR TX TO COMPLETE.
                                       BCC     50$              ;GO REPORT ERROR IF TX DID NOT COMPLETE.
                                ;+
                                ; Read three characters, to cause the XON to be generated.
                                ;-
                                        INC     ERRNBR          ;SET ERROR NUMBER TO 5707.
                                        MOV     #3,R0           ;SET THE READ COUNT TO 3.
                                       JSR     PC,READBX        ;READ 3 CHARACTERS FROM THE FIFO.
                                       BCC     50$              ;GO REPORT ERROR IF FIFO EMPTY.
                                        INC     ERRNBR          ;SET ERROR NUMBER TO 5708.
                                        TST     R1              ;CHECK IF AN XON WAS FOUND.
                                       BNE     40$              ;GO REPORT ERROR IF AN XON WAS FOUND.
                                ;+
                                ; Transmit 62 characters to bracket the XON and fill the fifo with 191 chars.
                                ;-
                                        MOV     #62.,R0         ;PASS LENGTH OF DATA PATTERN.
                                        MOV     R5,R1           ;PASS THE LINE NUMBER.
                                        INC     ERRNBR          ;SET ERROR NUMBER TO 5709.
                                       JSR     PC,TXDATP        ;TRANSMIT DATA PATTERN.
                                       BCC     50$              ;EXIT IF ERROR FOUND DURING TX.

                                    ;+
                                    ; Wait for DMA to complete, then wait for the last character to arrive in
                                    ; the fifo.
                                    ;-
                                        INC     ERRNBR          ;SET ERROR NUMBER TO 5710.
                                       JSR     PC,WAITTX        ;WAIT FOR TX TO COMPLETE.
                                       BCC     50$              ;GO REPORT ERROR IF TX FAILED TO COMPLETE.
                                ;+
                                ; Read the first 126 characters.
```

```
6363                                    ; Read the next 4 characters and check if they are in the following order
6364                                    ; NULL, XOFF, XON, NULL.
6365                                    ;-
6366 026176  005267  155704                     INC     ERRNBR          ;SET ERROR NUMBER TO 5711.
6367 026202  012700  000176                     MOV     #126.,R0        ;SET UP READ COUNTER.
6368 026206  004767  167076                    JSR     PC,READBX       ;READ THE FIRST 126 CHARS.
6369 026212  103056                           BCC     50$             ;GO REPORT THE ERROR IF FIFO EMPTY.
6370 026214  005267  155666                     INC     ERRNBR          ;SET ERROR NUMBER TO 5712.
6371 026220  005701                            TST     R1              ;CHECK IF AN XON WAS FOUND.
6372 026222  001050                           BNE     40$             ;GO REPORT ERROR IF AN XON WAS FOUND.
6373 026224  005267  155656                     INC     ERRNBR          ;SET ERROR NUMBER TO 5713.
6374 026230  012701  006744                     MOV     #EM5402,R1      ;PASS THE MESSAGE TO BE REPORTED.
6375 026234  016703  154004                     MOV     RBUFA,R3        ;GET THE RECEIVER BUFFER ADDRESS.
6376 026240  011302                            MOV     (R3),R2         ;READ THE NULL CHARACTER FROM THE FIFO.
6377 026242  120227  000000                     CMPB    R2,#000         ;CHECK IF IT IS A NULL CHARACTER.
6378 026246  001036                           BNE     40$             ;GO REPORT THE ERROR IF NOT THE SAME.
6379 026250  005267  155632                     INC     ERRNBR          ;SET ERROR NUMBER TO 5714.
6380 026254  011302                            MOV     (R3),R2         ;READ THE XOFF FROM THE FIFO.
6381 026256  120227  000023                     CMPB    R2,#23          ;CHECK IF THE READ CHAR IS AN XOFF.
6382 026262  001030                           BNE     40$             ;GO REPORT THE ERROR IF NOT THE SAME.
6383 026264  011302                            MOV     (R3),R2         ;READ THE XON FROM THE FIFO.
6384 026266  005267  155614                     INC     ERRNBR          ;SET ERROR NUMBER TO 5715.
6385 026272  120227  000021                     CMPB    R2,#21          ;CHECK IF THE READ CHARACTER IS AN XON.
6386 026276  001022                           BNE     40$             ;GO REPORT THE ERROR IF NOT THE SAME.
6387 026300  005267  155602                     INC     ERRNBR          ;SET ERROR NUMBER TO 5716.
6388 026304  011302                            MOV     (R3),R2         ;READ THE NULL CHARACTER FROM THE FIFO.
6389 026306  120227  000000                     CMPB    R2,#000         ;CHECK IF IT IS A NULL CHARACTER.
6390 026312  001014                           BNE     40$             ;GO REPORT THE ERROR IF NOT THE SAME.
6391
6392                                    ;+
6393                                    ; Read the remaining characters from the fifo.
6394                                    ;-
6395 026314  012700  000075     6$:             MOV     #61.,R0         ;SET UP READ COUNTER.
6396 026320  005267  155562                     INC     ERRNBR          ;SET ERROR NUMBER TO 5717.
6397 026324  004767  166760                    JSR     PC,READBX       ;READ THE FIRST 61 CHARS.
6398 026330  103007                           BCC     50$             ;GO REPORT THE ERROR IF FIFO EMPTY.
6399 026332  005267  155550                     INC     ERRNBR          ;SET ERROR NUMBER TO 5718.
6400 026336  005701                            TST     R1              ;CHECK IF AN XON WAS FOUND.
6401 026340  001001                           BNE     40$             ;GO REPORT ERROR IF AN XON WAS FOUND.
6402 026342  000404                            BR      60$             ;EXIT THE TEST.
6403 026344                             40$:    ERROR                   ;                >>>>> ERROR <<<<<
     026344  104460                                                                            TRAP    C$ERROR
6404 026346  000402                           BR      60$             ;EXIT THE TEST.
6405
6406 026350  004767  167426     50$:    JSR     PC,TSABRT       ;REPORT TEST ABORTED.  ERROR # INDICATES FAULT.
6407 026354  005067  153704     60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6408
6409 026360                            ENDTST
     026360                                                                          L10036:
     026360  104401                                                                         TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - BREAKB -
                                ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;*                      - BREAK generation test -
                                ;*      This test verifies that all serial transmit lines can generate a break
                                ;*      by setting the brk bit in the associated LNCTRL register.
                                ;*      Use of the internal loopback feature of the DUARTS is made to minimise
                                ;*      any external effects caused on the serial lines by this test.
                                ;*      Framing error detection is used to indicate the presence of a break,
                                ;*      by setting the appropriate bit in the RBUF register.
                                ;---------------------------------------------------------------------------------------

026362                                  BGNTST
026362                                                                  T13::

026362  012767  177777  153674          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
        000015                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026370  012767  000015  153670          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (64)
026376  012767  000001  155500          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026404  012767  014401  155474          MOV     #6401.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
026412  012767  007156  155470          MOV     #EM6401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
                                ;+
                                ; 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 >>>>> 6401 <<<<<.
                                ;-
026420  004767  165062                  JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
026424  103165                          BCC     60$             ;EXIT TEST IF FATAL ERROR FOUND.
                                ;+
                                ; Set up Device Under Test (DUT) to:
                                ;    Disable transmission and reception interrupts.
                                ;    Delay for 10 milli-seconds to allow time to clear any breaks.
                                ;-
026426  004767  167674                  JSR     PC,TXIE0        ;DISABLE TRANSMISSION INTERRUPTS.
026432  004767  167046                  JSR     PC,RXIE0        ;DISABLE RECEPTION INTERRUPTS.
026436  012705  000377                  MOV     #MAPLNS,R5      ;PASS ACTIVE LINE BIT MAP.
026442  012700  000200                  MOV     #200,R0         ;PASS INTERNAL LOOPBACK MODE.
026446  004767  170240                  JSR     PC,WTWLNC       ;SELECT INTERNAL LOOPBACK,DISABLE DMA.
026452  012704  000012                  MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
026456  004767  165236                  JSR     PC,DELAY        ;DELAY TO ALLOW ANY BREAKS TO BE CLEARED.
                                ;+
                                ; Set up transmission an reception parameters for all lines.
                                ; 9600 baud,8 char,1 stopbit,no parity.
                                ;-
026462  012700  156430                  MOV     #156430,R0      ;SET UP BAUD RATE,ETC.
026466  004767  170250                  JSR     PC,WTWLPR       ;SET COMMUNICATION PARAMETERS ON ALL LINES.
                                ;+
                                ; Enable transmitters on all active lines.
                                ;-
026472  016705  153536                  MOV     ACTLNS,R5       ;PASS ACTIVE LINE BIT MAP.
026476  004767  167530                  JSR     PC,TXENBL       ;ENABLE TRANSMISSIONS ON ALL LINES.
                                ;+
                                ; Purge the FIFO of any unwanted characters.
                                ; This routine reports errors with numbers >>>>> 6402 thru 6404 <<<<<.
                                ;-
026502  005267  155400                  INC     ERRNBR          ;SET ERROR NUMBER TO 6402.
026506  004767  166404                  JSR     PC,PUFIFR       ;PURGE FIFO.
026512  103132                          BCC     60$             ;ABORT TEST IF FIFO WILL NOT CLEAR.
```

```
                                        ;+
                                        ; Verify break generation on individual lines.
                                        ;  Clear breaks on all lines.
                                        ;  Delay for 10 milli-seconds to allow time for any breaks to be cleared.
                                        ;  Select line,set break bit in LNCTRL register.
                                        ;  Test for a character in the FIFO with frame error.
                                        ;-
026514  005002                    2$:      CLR     R2              ;CLEAR LINE COUNTER.
026516  012703  000001                     MOV     #1,R3           ;SET UP ACTIVE LINE BIT MASK.
026522  030367  153506             4$:      BIT     R3,ACTLNS       ;CHECK IF THIS LINE IS ACTIVE.
026526  001434                             BEQ     8$              ;GO SELECT NEXT LINE IF THIS ONE IS INACTIVE.
026530  012700  000200                     MOV     #200,R0         ;SET UP PARAMETER TO CLEAR BREAK BITS.
026534  004767  170152                     JSR     PC,WTWLNC       ;CLEAR BREAK BIT,RESELECT INTERNAL LOOPBACK.
026540  012704  000012                     MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
026544  004767  165150                     JSR     PC,DELAY        ;DELAY TO ALLOW BREAKS TO BE CLEARED.
                                        ;+
                                        ; Set break bit on selected line.
                                        ;  Set up parameters to test for the frame error bit set in RBUF.
                                        ;  Time-out = 5 milli seconds.
                                        ;  Call routine to check for condition found.
                                        ;-
026550  010305                    6$:      MOV     R3,R5           ;COPY ACTIVE LINE BIT MASK.
026552  012700  000214                     MOV     #214,R0         ;SET BREAK,RESELECT LOOPBACK,ENABLE RECEPTION.
026556  004767  170130                     JSR     PC,WTWLNC       ;SET BREAK ON SELECTED LINE.
                                        ;+
                                        ; Delay for 5 ms to allow time for break to be generated and received.
                                        ; Verify reception of a character with frame error bit set.
                                        ;-
026562  012704  000005                     MOV     #5.,R4          ;SET DELAY VALUE TO 5 MILLI SECS.
026566  004767  165126                     JSR     PC,DELAY        ;ALLOW TIME FOR CHARACTER RECEPTION.
026572  017700  153446                     MOV     @RBUFA,R0       ;GET CHARACTER FROM RBUF REGISTER.
026576  032700  020000                     BIT     #BIT13,R0       ;CHECK FOR FRAME ERROR BIT.
026602  001006                             BNE     8$              ;SKIP ERROR REPORT IF SET.
026604  012701  007205                     MOV     #EM6402,R1      ;SELECT MESSAGE TO BE PRINTED.
                                    ;REPORT ERROR"BREAK NOT RECEIVED ON LINE #nn"
026610                                      ERRDF   6405,EM6401,ER6401 ;           >>>>> ERROR #6405 <<<<<.
026610  104455                                                                              TRAP    C$ERDF
026612  014405                                                                            .WORD   6405
026614  007156                                                                            .WORD   EM6401
026616  011742                                                                            .WORD   ER6401
026620  006303                    8$:      ASL     R3              ;SHIFT BIT MASK FOR NEXT LINE.
026622  005202                             INC     R2              ;NEXT LINE
026624  020227  000010                     CMP     R2,#NUMLNS      ;CHECK FOR MAX LINE COUNT.
026630  001334                             BNE     4$              ;IF <>,LOOP TO CHECK NEXT LINE
                                        ;+
                                        ; Verify break generation on all lines simultaneously.
                                        ;  Clear breaks on all lines.
                                        ;  Delay for 10 milli-seconds to allow time for any breaks to be cleared.
                                        ;  Purge the FIFO.
                                        ;  Set break bit in LNCTRL registers on all active lines.
                                        ;  Test for characters in the FIFO with frame error.
                                        ;-
026632  012705  000377                     MOV     #MAPLNS,R5      ;SET UP LINE TO CLEAR BREAKS ON.
026636  012700  000200                     MOV     #200,R0         ;SET UP PARAMETER TO CLEAR BREAK BITS.
026642  004767  170044                     JSR     PC,WTWLNC       ;CLEAR BREAK BIT,RESELECT INTERNAL LOOPBACK.
026646  012704  000012                     MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
026652  004767  165042                     JSR     PC,DELAY        ;DELAY TO ALLOW BREAKS TO BE CLEARED.
```

```
                                        ;+
                                        ; Purge the FIFO of unwanted characters.
                                        ;-
026656  004767  166152                      JSR     PC,PUFIFO       ;PURGE FIFO.
026662  103044                            BCC     50$             ;GO REPORT ERROR IF FAILED TO CLEAN_OUT FIFO.
                                      ;+
                                      ; Set up parameters for setting the break bit on all active lines.
                                      ; Then call routine to do it.
                                      ;-
026664  016705  153344          10$:     MOV     ACTLNS,R5       ;SET UP ACTIVE LINE BIT MASK.
026670  012700  000214                   MOV     #214,R0         ;SET BREAK,RESELECT LOOPBACK,ENABLE RECEPTION.
026674  004767  170012                  JSR     PC,WTWLNC       ;SET BREAK ON SELECTED LINES.
                                        ;+
                                        ;  Delay for 10 milli seconds,to allow time for reception.
                                        ;  Test for characters in fifo with frame error bit set.
                                        ;-
026700  012704  000012                    MOV     #10.,R4         ;SET DELAY VALUE TO 10 MILLI SECS.
026704  004767  165010                  JSR     PC,DELAY        ;ALLOW TIME FOR CHARACTER RECEPTION.
026710  010502                           MOV     R5,R2           ;COPY ACTIVE LINE BIT MAP.
026712  004767  165424                  JSR     PC,MAPCNT       ;COUNT THE NUMBER OF LINES AVAILABLE.
026716  017701  153322          12$:     MOV     @RBUFA,R1       ;GET CHARACTER FROM RBUF REGISTER.
026722  100011                          BPL     14$             ;BRANCH IF DATA_VALID NOT SET.
026724  032701  020000                   BIT     #BIT13,R1       ;CHECK FOR FRAME ERROR BIT.
026730  001406                           BEQ     14$             ;DO NOT CLR FLG FOR THIS LINE IF FRAME BIT CLR.
026732  000301                           SWAB    R1              ;GET LINE NUMBER IN LOW BYTE.
026734  042701  177400                   BIC     #177400,R1      ;CLEAR EVERTHING BUT THE LINE NUMBER.
026740  004767  165350                  JSR     PC,LINBIT       ;CALC BIT MASK FROM LINE NUMBER.
026744  040005                           BIC     R0,R5           ;CLEAR LINE FLAG.
026746  005302                  14$:     DEC     R2              ;DECREMENT THE LINE NUMBER COUNTER.
026750  001362                          BNE     12$             ;LOOP TO GET THE NEXT CHARACTER.
026752  005705                           TST     R5              ;CHECK IF ANY BREAKS NOT RECEIVED.
026754  001411                          BEQ     60$             ;EXIT TEST IF ALL CLEAR.
026756  012701  007205                   MOV     #EM6402,R1      ;SELECT MESSAGE TO BE PRINTED.
                                  ;REPORT ERROR"BREAK NOT RECEIVED ON LINE #nn".
026762                                    ERRDF   6406,EM6401,ER6401;             >>>>> ERROR #6407 <<<<<.
026762  104455                                                                          TRAP    C$ERDF
026764  014406                                                                        .WORD   6406
026766  007156                                                                        .WORD   EM6401
026770  011742                                                                        .WORD   ER6401
026772  000402                          BR      60$             ;EXIT THE TEST.

026774  004767  167002          50$:     JSR     PC,TSABRT       ;ABORT THE TEST.
027000  005067  153260          60$:      CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
027004                                    ENDTST
027004                                                                      L10037:
027004  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - NORERR -
                                ;+*****************************************************************************
                                ;*                        - NO OVERRUN ERROR TEST -
                                ;*
                                ;*      This test verifies that the DUT will not report data overrun
                                ;*      errors when they do not occurr.
                                ;*      This test puts 256 characters in the DUT FIFO plus 4 in each active
                                ;*      UART and verifies that no overrun errors are reported.
                                ;*      Any BMP code found will invalidate the test and cause it to be aborted.
                                ;*      However the BMP code will be placed on the BMP code queue, to be
                                ;*      reported later.
                                ;*
                                ;--*****************************************************************************

027006                                  BGNTST
027006                                                                  T14::
        000016                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027006  012767  000016  153252   MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (66)
027014  012767  177777  153242   MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
027022  012767  000001  155054   MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027030  012767  014711  155050   MOV    #6601.,ERRNBR   ;SET ERROR NUMBER TO 6601.
027036  012767  007246  155044   MOV    #EM6601,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; 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 >>>>> 6601 <<<<<.
                                ;-
027044  004767  164436          JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
027050  103402                  BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
027052  000167  000422          JMP     60$             ;EXIT THE TEST, FATAL ERROR WAS FOUND.
                                ;+
                                ; Find an active line on which to perform the test.
                                ; Initialize the 256 byte data pattern.
                                ;-
027056  004767  165052          JSR     PC,FINACT       ;FIND AN ACTIVE LINE.
027062  103402                  BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
027064  000167  000410          JMP     60$             ;EXIT THE TEST, FATAL ERROR WAS FOUND.
027070  004767  165120          JSR     PC,INDATP       ;INITIALISE DATA PATTERN.
                                ;+
                                ; Transmit a 265 character data pattern using DMA, on a single channel
                                ; at 38.4k baud, 8 bits per character, odd parity, 2 stop bits.
                                ;-
                                    ;+
                                    ; Set internal loopback on the selected line.
                                    ; Transmit the data pattern on the first available active line.
                                    ;-
027074  005267  155006              INC     ERRNBR          ;SET THE ERROR REPORT NUMBER TO 6602.
027100  012700  000204              MOV     #204,R0         ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
027104  004767  167602              JSR     PC,WTWLNC       ;INITILAISE THE LINE CONTROL REGISTER.
027110  012700  177670              MOV     #177670,R0      ;PASS THE LPR CONTENTS.
027114  004767  167622              JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
027120  012704  000012              MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
027124  004767  164570              JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
027130  012702  002704              MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
027134  012703  000400              MOV     #BUFMID-BUFBAS,R3 ;PASS THE LENGTH OF THE DATA PATTERN.
027140  004767  164614              JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
027144  103153                      BCC     50$             ;EXIT IF ERROR FOUND DURING DMA TX.
```

```
                                        ;+
                                        ; Wait for DMA to complete, then wait for the last character to arrive in
                                        ; the fifo.
                                        ;-
027146  005267  154734                      INC     ERRNBR          ;SET ERROR NUMBER TO 6603.
027152  012701  170536                      MOV     #170536,R1      ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
027156  016702  153060                      MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
027162  004767  167410                    JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
027166  103142                            BCC     50$             ;ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
027170  012704  000005                      MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
027174  004767  164520                    JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.

                                    ;+
                                    ; Transmit 4 characters on each active line.
                                    ;-
027200  016705  153030                      MOV     ACTLNS,R5       ;ALTER PARAMETERS FOR ALL ACTIVE LINES.
027204  012700  000204                      MOV     #204,R0         ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
027210  004767  167476                    JSR     PC,WTWLNC       ;INITILAISE THE LINE CONTROL REGISTER.
027214  012700  177670                      MOV     #177670,R0      ;PASS THE LPR CONTENTS.
027220  004767  167516                    JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
027224  012704  000012                      MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
027230  004767  164464                    JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.

027234  012702  002704                      MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
027240  012703  000004                      MOV     #4,R3           ;PASS THE LENGTH OF THE DATA PATTERN.
027244  005001                              CLR     R1              ;CLEAR THE LINE COUNTER.
027246  005267  154634                      INC     ERRNBR          ;SET ERROR NUMBER TO 6604.
027252  010100                    2$:       MOV     R1,R0
027254  006300                              ASL     R0              ;CALCULATE THE LINE OFFSET FROM THE LINE #.
027256  036067  002370  152750              BIT     BITTBL(R0),ACTLNS  ;TEST FOR THIS LINE BEING ACTIVE.
027264  001403                            BEQ     4$              ;SKIP THE TX ON THIS LINE IF IT IS NOT ACTIVE.
027266  004767  164466                    JSR     PC,DODMA        ;TRANSMIT THE 5 CHAR DATA PATTERN.
027272  103100                            BCC     50$             ;ABORT IF ERROR FOUND DURING DMA TX.
027274  005201                    4$:       INC     R1              ;INCREMENT THE LINE COUNTER.
027276  020127  000010                      CMP     R1,#NUMLNS      ;TEST FOR ALL POSSIBLE LINES HANDLED
027302  002763                            BLT     2$              ;LOOP IF NOT ALL LINES HANDLED.

027304  005267  154576                      INC     ERRNBR          ;SET ERROR NUMBER TO 6605.
027310  012701  170040                      MOV     #170040,R1      ;PASS TIME-OUT VALUE OF 32 MILLI SECS.
027314  016702  152722                      MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
027320  004767  167252                    JSR     PC,WAIBIS       ;WAIT FOR A DMA TO COMPLETE, TX_ACTION SET.
027324  103063                            BCC     50$             ;ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
027326  012704  000005                      MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
027332  004767  164362                    JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
                                    ;+
                                    ; Read the FIFO checking for overrun errors.  Report errors if found.
                                    ; Abort the test if a BMP code was found in the fifo.
                                    ;-
027336  016702  152672                      MOV     ACTLNS,R2
027342  004767  164774                    JSR     PC,MAPCNT       ;GET THE NUMBER OF ACTIVE LINES.
027346  006302                              ASL     R2
027350  006302                              ASL     R2              ;MULTIPLY NUMBER OF ACTIVE LINES BY 4.
027352  012705  000400                      MOV     #256.,R5
027356  060205                              ADD     R2,R5           ;CALCULATE NUMBER OF CHARACTERS TO RX.
027360  005004                              CLR     R4              ;CLEAR THE CHARACTER COUNTER.
027362  012767  014716  154516    6$:       MOV     #6606.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
027370  017702  152650                      MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE FIFO.
```

```
                                        BPL     10$             ;EXIT THE READ LOOP IF THE FIFO IS EMPTY.
                                    ;+
                                    ; Check if the read character is a BMP code.
                                    ;  If it is a BMP code save it on the queue to be reported later, and
                                    ;  abort the test.
                                    ;-
                                        JSR     PC,CHKBMP       ;CHECK IF CHARACTER IS A BMP CODE.
                                        BCC     8$              ;BRANCH IF NOT A BMP CODE.
                                        ERROR                   ;               >>>>> ERROR #6606 <<<<<.
                                                                                        TRAP    C$ERROR
                                        BR      60$             ;EXIT THIS TEST.

                                8$:      INC    ERRNBR          ;SET ERROR NUMBER TO 6607.
                                         INC    R4              ;COUNT THIS CHARACTER.
                                        CMP     R4,R5           ;COMPARE # OF CHARS WITH MAX # OF CHARS.
                                         BGT    50$             ;ABORT TEST IF TOO MANY VALID CHARS READ.
                                        BIT     #BIT14,R2       ;TEST THE OVERRUN BIT OF THE READ CHAR.
                                         BEQ    6$              ;LOOP TO READ THE NEXT CHAR IF NO ERROR.
                                         INC    ERRNBR          ;SET ERROR NUMBER TO 6608.
                                         MOV    #ER7801,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                         MOV    #EM6602,R1      ;PASS THE MESSAGE TO BE REPORTED.
                                         MOV    R2,R3
                                         SWAB   R3
                                         BIC    #177760,R3      ;GET FAILING LINE NUMBER.
                                    ;REPORT "OVERRUN ERROR REPORTED WHEN NONE FORCED, ON LINE nn ..."
                                        ERROR                   ;               >>>>> ERROR  #6608 <<<<<.
                                                                                        TRAP    C$ERROR
                                        BR      6$              ;LOOP TO READ THE NEXT CHAR.

                                10$:     MOV    #6609.,ERRNBR   ;SET ERROR NUMBER TO 6609.
                                         CMP    R4,R5           ;COMPARE NUMBER OF CHARS READ WITH EXPECTED.
                                        BEQ     60$             ;EXIT TEST WITHOUT ABORT IF CORRECT # OF CHARS.

                                50$:    JSR     PC,TSABRT       ;ABORT THE TEST, NON-RELATED TEST ERROR FOUND.
                                60$:     CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
                                        ENDTST
                                                                                L10040:
                                                                                        TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - ORERR -
                                ;+*****************************************************************************
                                ;*                          - OVERRUN ERROR TEST -
                                ;*
                                ;*      This test verifies that the DUT will report data overrun errors when
                                ;*      they occurr.
                                ;*      This test puts 256 characters in the DUT FIFO plus 5 in each active
                                ;*      UART and verifies that overrun errors are reported on all active lines.
                                ;*      Any BMP code found will invalidate the test and cause it to be aborted.
                                ;*      However the BMP code will be placed on the BMP code queue, to be
                                ;*      reported later.
                                ;*
                                ;--*****************************************************************************

027506                                  BGNTST
027506                                                                          T15::
        000017                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027506  012767  000017  152552   MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (67)
027514  012767  177777  152542   MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
027522  012767  000001  154354   MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027530  012767  015055  154350   MOV    #6701.,ERRNBR   ;SET ERROR NUMBER TO 6701.
027536  012767  007346  154344   MOV    #EM6701,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; 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 >>>>> 6701 <<<<<.
                                ;-
027544  004767  163736          JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
027550  103402                  BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
027552  000167  000650          JMP     60$             ;EXIT THE TEST, FATAL ERROR WAS FOUND.
                                ;+
                                ; Find an active line on which to perform the test.
                                ; Initialize the 256 byte data pattern.
                                ;-
027556  004767  164352          JSR     PC,FINACT       ;FIND AN ACTIVE LINE.
027562  103402                  BCS     .+6             ;IF ACTIVE LINE IS FOUND, DON'T ABORT TEST.
027564  000167  000636          JMP     60$             ;ABORT THE TEST, NO ACTIVE LINES WERE FOUND.
027570  004767  164420          JSR     PC,INDATP       ;INITIALISE DATA PATTERN.
                                ;+
                                ; Transmit a 265 character data pattern using DMA, on a single channel
                                ; at 38.4k baud, 8 bits per character, odd parity, 2 stop bits.
                                ;-
                                    ;+
                                    ; Set internal loopback on the selected line.
                                    ; Transmit the data pattern on the first available active line.
                                    ;-
027574  005267  154306              INC     ERRNBR          ;SET ERROR NUMBER TO 6702.
027600  012700  000204              MOV     #204,R0         ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
027604  004767  167102              JSR     PC,WTWLNC       ;INITILAISE THE LINE CONTROL REGISTER.
027610  012700  177670              MOV     #177670,R0      ;PASS THE LPR CONTENTS.
027614  004767  167122              JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
027620  012704  000012              MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
027624  004767  164070              JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
027630  012702  002704              MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
027634  012703  000400              MOV     #BUFMID-BUFBAS,R3 ;PASS THE LENGTH OF THE DATA PATTERN.
027640  004767  164114              JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERN.
027644  103402                      BCS     .+6             ;IF NO ERROR FOUND DURING DMA TX, DON'T ABORT.
```

```
 6765 027646  000167  000550                  JMP     50$             ;ABORT TEST, ERROR FOUND DURING DMA TX.
 6766                                        ;+
 6767                                        ; Wait for DMA to complete, then wait for the last character to arrive in
 6768                                        ; the fifo.
 6769                                        ;-
 6770 027652  005267  154230                   INC     ERRNBR          ;SET ERROR NUMBER TO 6703.
 6771 027656  012701  170536                   MOV     #170536,R1      ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
 6772 027662  016702  152354                   MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
 6773 027666  004767  166704                  JSR     PC,WAIBIS       ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
 6774 027672  103402                          BCS     .+6             ;IF NO TIME-OUT ON DMA COMPLETION, DON'T ABORT.
 6775 027674  000167  000522                  JMP     50$             ;ABORT TEST, TIME-OUT ON DMA COMPLETION.
 6776 027700  012704  000005                   MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
 6777 027704  004767  164010                  JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
 6778                                      ;+
 6779                                      ; Transmit 5 characters on each active line.
 6780                                      ;-
 6781 027710  016705  152320                   MOV     ACTLNS,R5       ;ALTER PARAMETERS FOR ALL ACTIVE LINES.
 6782 027714  012700  000204                   MOV     #204,R0         ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
 6783 027720  004767  166766                  JSR     PC,WTWLNC       ;INITILAISE THE LINE CONTROL REGISTER.
 6784 027724  012700  177670                   MOV     #177670,R0      ;PASS THE LPR CONTENTS.
 6785 027730  004767  167006                  JSR     PC,WTWLPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
 6786 027734  012704  000012                   MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
 6787 027740  004767  163754                  JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
 6788
 6789 027744  012702  002704                   MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
 6790 027750  012703  000005                   MOV     #5,R3           ;PASS THE LENGTH OF THE DATA PATTERN.
 6791 027754  005001                           CLR     R1              ;CLEAR THE LINE COUNTER.
 6792 027756  005267  154124                   INC     ERRNBR          ;SET ERROR NUMBER TO 6704.
 6793 027762  010100                  2$:      MOV     R1,R0
 6794 027764  006300                           ASL     R0              ;CALCULATE LINE OFFSET FROM THE LINE #.
 6795 027766  036067  002370  152240           BIT     BITTBL(R0),ACTLNS  ;TEST FOR THIS LINE BEING ACTIVE.
 6796 027774  001405                          BEQ     4$              ;SKIP THE TX ON THIS LINE IF IT IS NOT ACTIVE.
 6797 027776  004767  163756                  JSR     PC,DODMA        ;TRANSMIT THE 5 CHAR DATA PATTERN.
 6798 030002  103402                          BCS     .+6             ;IF NO TIME-OUT ON DMA COMPLETION, DON'T ABORT.
 6799 030004  000167  000412                  JMP     50$             ;ABORT TEST, TIME-OUT ON DMA COMPLETION.
 6800 030010  005201                  4$:      INC     R1              ;INCREMENT THE LINE NUMBER COUNTER.
 6801 030012  020127  000010                   CMP     R1,#NUMLNS      ;TEST FOR ALL POSSIBLE LINES HANDLED
 6802 030016  002761                          BLT     2$              ;LOOP IF NOT ALL LINES HANDLED.
 6803
 6804 030020  005267  154062                   INC     ERRNBR          ;SET ERROR NUMBER TO 6705.
 6805 030024  012701  170040                   MOV     #170040,R1      ;PASS TIME-OUT VALUE OF 32 MILLI SECS.
 6806 030030  016702  152206                   MOV     CSRA,R2         ;PASS THE ADDRESS OF THE CSR.
 6807 030034  004767  166536                  JSR     PC,WAIBIS       ;WAIT FOR A DMA TO COMPLETE, TX_ACTION SET.
 6808 030040  103170                          BCC     50$             ;ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
 6809 030042  012704  000005                   MOV     #5,R4           ;PASS DELAY OF 5 MILLI SECS.
 6810 030046  004767  163646                  JSR     PC,DELAY        ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
 6811                                      ;+
 6812                                      ; Read 256 chars from the FIFO checking for BMP codes.
 6813                                      ; Abort the test if a BMP code was found in the fifo.
 6814                                      ;-
 6815 030052  012704  000400                   MOV     #256.,R4        ;SET UP THE CHARACTER COUNTER.
 6816 030056  012767  015062  154022  6$:      MOV     #6706.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
 6817 030064  017702  152154                   MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE FIFO.
 6818 030070  100154                          BPL     50$             ;ABORT THE TEST IF DATA.VALID IS CLEAR.
 6819 030072  005267  154010                   INC     ERRNBR          ;SET ERROR NUMBER TO 6707.
 6820 030076  004767  163304                  JSR     PC,CHKBMP       ;CHECK IF CHARACTER IS A BMP CODE.
 6821 030102  103545                          BCS     24$             ;REPORT ERROR AND ABORT TEST IF A BMP CODE.
```

```
 6822 030104  005304                          DEC     R4              ;COUNT THIS CHARACTER.
 6823 030106  001363                          BNE     6$              ;LOOP IF NOT 256 CHARS READ FROM FIFO.
 6824                                 ;+
 6825                                 ; Read the remaining and verify 1 overrun plus 1 char from each line.
 6826                                 ;-
 6827 030110  005004                          CLR     R4              ;CLEAR THE OVERRUN ERROR FLAGS.
 6828 030112  012700  003744                  MOV     #RXCNTB,R0
 6829 030116  004767  163406                  JSR     PC,CLR16W       ;CLEAR RX CHAR COUNT TABLE.
 6830 030122  012767  015064  153756  8$:    MOV     #6708.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
 6831 030130  017702  152110                  MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE FIFO.
 6832 030134  100047                          BPL     14$             ;GO ANALYZE THE RESULTS IF ALL CHARS READ.
 6833 030136  004767  163244                  JSR     PC,CHKBMP       ;CHECK IF CHAR IS A BMP CODE.
 6834 030142  103525                          BCS     24$             ;REPORT ERROR AND ABORT TEST IF A BMP CODE.
 6835 030144  005267  153736                  INC     ERRNBR          ;SET ERROR NUMBER TO 6709.
 6836 030150  010200                          MOV     R2,R0
 6837 030152  000300                          SWAB    R0
 6838 030154  042700  177760                  BIC     #177760,R0      ;CALCULATE THE LINE NUMBER OF THE CHAR.
 6839 030160  006300                          ASL     R0              ;FORM WORD TABLE OFFSET FOR TABLE ACCESS.
 6840 030162  042702  007400                  BIC     #7400,R2        ;REMOVE LINE NUMBER FROM THE READ CHAR.
 6841 030166  036067  002370  152040          BIT     BITTBL(R0),ACTLNS       ;TEST FOR ACTIVE LINE.
 6842 030174  001512                          BEQ     50$             ;ABORT TEST IF FOR INACTIVE LINE.
 6843 030176  005267  153704                  INC     ERRNBR          ;SET ERROR NUMBER TO 6710.
 6844 030202  005760  003744                  TST     RXCNTB(R0)      ;CHECK THE RX CHAR COUNTER FOR THIS LINE.
 6845 030206  001006                          BNE     10$             ;IS THIS FIRST CHAR ON LINE?
 6846 030210  020227  140000                  CMP     R2,#140000      ;YES, TEST FOR NULL CHAR WITH OVERRUN.
 6847 030214  001414                          BEQ     12$             ;IS CHAR A NULL?
 6848 030216  056004  002370                  BIS     BITTBL(R0),R4   ;NO, SET THE OVERRUN BIT ERROR FLAG FOR LINE.
 6849 030222  000411                          BR      12$             ;GO COUNT THE CHAR AND CONTINUE.
 6850 030224  026027  003744  000004  10$:    CMP     RXCNTB(R0),#4
 6851 030232  002073                          BGE     50$             ;5TH CHAR ON THIS LINE?  YES, ABORT.
 6852 030234  032702  040000                  BIT     #BIT14,R2       ;NO, CHECK OVERRUN BIT.
 6853 030240  001402                          BEQ     12$             ;IS OVERRUN BIT CLEAR?  YES, GO COUNT CHAR.
 6854 030242  056004  002370                  BIS     BITTBL(R0),R4   ;NO, SET THE OVERRUN BIT ERROR FLAG FOR LINE.
 6855 030246  005260  003744          12$:    INC     RXCNTB(R0)      ;COUNT THIS CHARACTER.
 6856 030252  000723                          BR      8$              ;LOOP UNTIL ALL CHARS ARE READ FROM FIFO.
 6857                                 ;+
 6858                                 ; Test for abort conditions.  Only none abort conditions are:
 6859                                 ;    1) 2 chars RXed on a line and no overrun error bit failure detected.
 6860                                 ;    2) 2 to 4 chars RXed on a line and an overrun bit failure detected.
 6861                                 ;-
 6862 030254  005001                  14$:    CLR     R1              ;INITIALIZE LINE LOOP, CLEAR LINE OFFSET.
 6863 030256  012767  015067  153622  16$:    MOV     #6711.,ERRNBR   ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
 6864 030264  036167  002370  151742          BIT     BITTBL(R1),ACTLNS
 6865 030272  001415                          BEQ     18$             ;LINE ACTIVE?  NO, NEXT LINE.
 6866 030274  026127  003744  000002          CMP     RXCNTB(R1),#2   ;YES.
 6867 030302  002447                          BLT     50$             ;FEWER THAN 2 CHARS RXED?  YES, ABORT.
 6868 030304  036104  002370                  BIT     BITTBL(R1),R4   ;NO.
 6869 030310  001006                          BNE     18$             ;OVERRUN BIT ERROR FLAG SET?  YES, NEXT LINE.
 6870 030312  005267  153570                  INC     ERRNBR          ;SET LINE NUMBER TO 6712.
 6871 030316  026127  003744  000002          CMP     RXCNTB(R1),#2
 6872 030324  001036                          BNE     50$             ;NOT 2 CHARS RXED?  YES, ABORT.  NO, NEXT LINE.
 6873 030326  062701  000002          18$:    ADD     #2,R1           ;SET LINE OFFSET TO THE NEXT LINE.
 6874 030332  020127  000020                  CMP     R1,#NUMLNS*2
 6875 030336  002747                          BLT     16$             ;ALL LINES DONE?  NO, LOOP.  YES, CONTINUE.
 6876                                 ;+
 6877                                 ; Check for overrun error bit failures, print error message if found.
 6878                                 ;-
```

HARDWARE TEST - ORERR -

```
6879 030340  012767  015071  153540          MOV     #6713.,ERRNBR   ;SET UP ERROR NUMBER.
6880 030346  005001                          CLR     R1              ;INITIALIZE LOOP.  CLEAR LINE OFFSET.
6881 030350  010102                  20$:    MOV     R1,R2           ;COPY THE LINE OFFSET.
6882 030352  036104  002370                  BIT     BITTBL(R1),R4   ;OVERRUN BIT FAILURE FLAGS ARE IN R4.
6883 030356  001411                          BEQ     22$             ;ERROR FLAG CLEAR?  YES, NEXT LINE.
6884 030360  010103                          MOV     R1,R3
6885 030362  006203                          ASR     R3              ;CALCULATE LINE NUMBER FROM LINE OFFSET.
6886 030364  012767  012040  153520          MOV     #ER7801,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6887 030372  012701  007371                  MOV     #EM6702,R1      ;PASS THE MESSAGE TO BE REPORTED.
6888                                  ;REPORT "OVERRUN ERROR NOT REPORTED CORRECTLY WHEN FORCED, ON LINE nn ..."
6889 030376                          ERROR                           ;               >>>>> ERROR  #6713 <<<<<.
     030376  104460                                                                  TRAP    C$ERROR
6890 030400  010201                          MOV     R2,R1           ;RESTORE THE LINE OFFSET THAT WAS DESTROYED.
6891 030402  046104  002370          22$:    BIC     BITTBL(R1),R4   ;CLEAR THE LINE ERROR FLAG WE JUST HANDLED.
6892 030406  001407                          BEQ     60$             ;ALL FAILURE BITS HANDLED?  YES, EXIT TEST.
6893 030410  062701  000002                  ADD     #2,R1           ;NO, INCREMENT THE LINE OFFSET.
6894 030414  000755                          BR      20$             ;LOOP TO HANDLE THE NEXT LINE.
6895
6896 030416                          24$:    ;REPORT "BMP CODE FOUND IN FIFO, TEST INVAILDATED."
6897 030416                          ERROR                           ;               >>>>> ERROR <<<<<.
     030416  104460                                                                  TRAP    C$ERROR
6898 030420  000402                          BR      60$             ;EXIT THIS TEST.
6899
6900 030422  004767  165354          50$:    JSR     PC,TSABRT       ;ABORT THE TEST.  ERROR # INDICATES FAULT TYPE.
6901 030426  005067  151632          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6902
6903 030432                          ENDTST
     030432                                                                  L10041:
     030432  104401                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - DTRMCS -
                                ;+*****************************************************************************
                                ;*              - Data Terminal Ready Modem Control Signal Test -
                                ;*
                                ;*      This test verifies that the DTR modem control signal is working
                                ;*      correctly.  It will only be performed if either 25 pin or staggered
                                ;*      loopback is specified.  This test uses the looped back signals RI
                                ;*      and DSR to test the DTR signal.  This test is performed on all
                                ;*      active lines.
                                ;*
                                ;--*****************************************************************************

030434                                  BGNTST
030434                                                                  T16::
                                ;+
                                ; Only perform this test if the DUT is in external or staggared loopback mode.
                                ;-
030434  032767  000002  151574          BIT     #BIT1,LOPBCK    ;CHECK TYPE OF LOOPBACK MODE SELECTED.
030442  001002                          BNE     2$
030444  000167  000456                  JMP     60$             ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
030450  000020                  2$:      TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
030450  012767  000020  151610          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (78)
030456  012767  177777  151600          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
030464  012767  000001  153412          MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
030472  012767  017171  153406          MOV     #7801.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
030500  012767  007446  153402          MOV     #EM7801,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Reset the DUT to a known state, remove status codes from the fifo.
                                ;  Clear TX and  RX interrupt enable bits.
                                ; This subroutine reports error >>>>> 7801 <<<<<.
                                ;-
030506  004767  162774                  JSR     PC,CLNRST       ;RESET THE DUT.
030512  103402                          BCS     4$
030514  000167  000406                  JMP     60$             ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; Set up the TX/RX associated line number table.
                                ;-
030520  004767  162326          4$:     JSR     PC,ASLNTL       ;SET UP THE ASSOCIATED LINE TABLES.
                                ;+
                                ; Set up a loop which handles one line per iteration.
                                ; This loop clears all the DTRs and then sets them individually and waits for
                                ;   a response on the associated RI and DSR signals.
                                ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                ;-
030524  005003                  6$:      CLR    R3              ;CLEAR THE LINE COUNTER.
030526  010300                          MOV     R3,R0
030530  006300                          ASL     R0
030532  036067  002370  151474          BIT     BITTBL(R0),ACTLNS
030540  001465                          BEQ     12$             ;DON'T TEST IF NOT ACTIVE LINE.
                                ;+
                                ; Clear all the DUT LNCTRL registers DTR bits.
                                ;-
030542  005000                          CLR     R0              ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
030544  012705  000377                  MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
030550  004767  166136                  JSR     PC,WTWLNC       ;CLEAR ALL THE DUT DTR BITS.
030554  012704  000074                  MOV     #60.,R4
030560  004767  163134                  JSR     PC,DELAY        ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.
```

```
                                        ;+
                                        ; Check that at least one of associated DSR or RI is clear and record states.
                                        ;-
030564  116304  004044                          MOVB    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
030570  010477  151446                          MOV     R4,@CSRA        ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
030574  017705  151450                          MOV     @STATA,R5       ;GET THE STATE OF THE ASSOCIATED DSR, RI BITS.
030600  012700  120000                          MOV     #BIT15!BIT13,R0
030604  040500                                  BIC     R5,R0           ;CHECK FOR BOTH DSR AND RI SET.
030606  001431                                BEQ     10$             ;GO REPORT DTR IS BAD IF BOTH ARE SET.
                                        ;+
                                        ; Set the DTR for the selected line and wait for either DSR or RI to set.
                                        ;-
030610  010377  151426                          MOV     R3,@CSRA        ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
030614  052777  001000  151430                  BIS     #BIT9,@LNCTRA   ;SET THE SELECTED LINE DTR.
030622  012701  150074                          MOV     #150074,R1      ;SPECIFY TO WAIT UP TO 60 MS FOR RI TO SET.
030626  032705  100000                          BIT     #BIT15,R5       ;CHECK PREVIOUS STATE OF DSR BIT.
030632  001002                                BNE     8$              ;GO USE RI IF DSR BIT WAS NOT CLEAR.
030634  012701  170074                          MOV     #170074,R1      ;SPECIFY TO WAIT UP TO 60 MS FOR DSR SET.
030640  016702  151404                  8$:     MOV     STATA,R2        ;SPECIFY TO LOOK IN STAT REG FOR BIT TO SET.
030644  010477  151372                          MOV     R4,@CSRA        ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
030650  004767  165722                        JSR     PC,WAIBIS       ;WAIT UP TO 60 MS FOR SIGNAL TO GO SET.
030654  103417                                BCS     12$             ;SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.
030656  017700  151366                          MOV     @STATA,R0       ;GET THE STATUS REGISTER CONTENTS.
030662  042700  057777                          BIC     #57777,R0       ;REMOVE ALL BUT THE DSR AND RI BITS.
030666  040500                                  BIC     R5,R0           ;TEST FOR SIGNAL ONCE CLEAR, BUT NOW SET.
030670  001011                                BNE     12$             ;GO LOOP IF SIGNAL HAS GONE FROM CLR TO SET.
030672                                  10$:    ;Report DTR MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
030672  012767  017172  153206                  MOV     #7802.,ERRNBR   ;SELECT THE ERROR NUMBER.
030700  012767  012040  153204                  MOV     #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
030706  012701  007501                          MOV     #EM7802,R1      ;SELECT THE ERROR MESSAGE.
030712                                          ERROR
030712  104460                                                                  TRAP    C$ERROR
030714  005203                          12$:     INC     R3              ;SELECT THE NEXT LINE NUMBER.
030716  020327  000010                          CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
030722  002701                                BLT     6$              ;LOOP IF NOT ALL LINES DONE.
                                        ;+
                                        ; Set up a loop which handles one line per iteration.
                                        ; This loop sets all the DTRs and then clears them individually and waits for
                                        ;   a response on the associated RI and DSR signals.
                                        ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                        ;-
030724  005003                                  CLR     R3              ;CLEAR THE LINE COUNTER.
030726  010300                          14$:     MOV     R3,R0
030730  006300                                  ASL     R0
030732  036067  002370  151274                  BIT     BITTBL(R0),ACTLNS
030740  001466                                BEQ     20$             ;DON'T TEST IF NOT ACTIVE LINE.
                                        ;+
                                        ; Set all the DUT LNCTRL registers DTR bits.
                                        ;-
030742  012700  001000                          MOV     #BIT9,R0        ;SPECIFY THAT DTR BITS ARE TO BE SET.
030746  012705  000377                          MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
030752  004767  165734                        JSR     PC,WTWLNC       ;SET ALL THE DUT DTR BITS.
030756  012704  000074                          MOV     #60.,R4
030762  004767  162732                        JSR     PC,DELAY        ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.
                                        ;+
                                        ; Check that at least one of associated DSR or RI is set and record states.
                                        ;-
```

```
7017 030766  116304  004044                    MOVB    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
7018 030772  010477  151244                    MOV     R4,@CSRA        ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
7019 030776  017705  151246                    MOV     @STATA,R5       ;GET THE STATE OF THE ASSOCIATED DSR, RI BITS.
7020 031002  010500                            MOV     R5,R0
7021 031004  042700  057777                    BIC     #57777,R0       ;CHECK FOR BOTH DSR AND RI CLEAR.
7022 031010  001431                          BEQ     18$             ;GO REPORT DTR IS BAD IF BOTH ARE CLEAR.
7023                                  ;+
7024                                  ; Clear the DTR for the selected line and wait for either DSR or RI to clear.
7025                                  ;-
7026 031012  010377  151224                    MOV     R3,@CSRA        ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
7027 031016  042777  001000  151226            BIC     #BIT9,@LNCTRA   ;CLEAR THE SELECTED LINE DTR.
7028 031024  012701  150074                    MOV     #150074,R1      ;SPECIFY TO WAIT UP TO 60 MS FOR RI TO CLEAR.
7029 031030  032705  100000                    BIT     #BIT15,R5       ;CHECK PREVIOUS STATE OF DSR BIT.
7030 031034  001402                          BEQ     16$             ;GO USE RI IF DSR BIT WAS NOT SET.
7031 031036  012701  170074                    MOV     #170074,R1      ;SPECIFY TO WAIT UP TO 60 MS FOR DSR CLEAR.
7032 031042  016702  151202          16$:      MOV     STATA,R2        ;SPECIFY TO LOOK IN STAT REG FOR BIT TO CLR.
7033 031046  010477  151170                    MOV     R4,@CSRA        ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
7034 031052  004767  165444                    JSR     PC,WAIBIC       ;WAIT UP TO 60 MS FOR SIGNAL TO GO CLEAR.
7035 031056  103417                          BCS     20$             ;SELECT NEXT LINE AND LOOP IF SIGNAL IS CLEAR.
7036 031060  017700  151164                    MOV     @STATA,R0       ;GET THE STATUS REGISTER CONTENTS.
7037 031064  042705  057777                    BIC     #57777,R5
7038 031070  040005                            BIC     R0,R5           ;TEST FOR SIGNAL ONCE SET, BUT NOW CLEAR.
7039 031072  001011                          BNE     20$             ;GO LOOP IF SIGNAL HAS GONE FROM SET TO CLR.
7040 031074                          18$:    ;Report DTR MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
7041 031074  012767  017173  153004            MOV     #7803.,ERRNBR   ;SELECT THE ERROR NUMBER.
7042 031102  012767  012040  153002            MOV     #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
7043 031110  012701  007501                    MOV     #EM7802,R1      ;SELECT THE ERROR MESSAGE.
7044 031114                                    ERROR
     031114  104460                                                          TRAP    C$ERROR
7045 031116  005203          20$:      INC     R3              ;SELECT THE NEXT LINE NUMBER.
7046 031120  020327  000010                    CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
7047 031124  002700                          BLT     14$             ;LOOP IF NOT ALL LINES DONE.
7048
7049 031126  005067  151132          60$:      CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7050
7051 031132                                    ENDTST
     031132                                                          L10042:
     031132  104401                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST          - RTSMCS -
                                ;++*************************************************************************
                                ;*              - Request to Send Modem Control Signal Test -
                                ;*
                                ;*      This test verifies that the RTS modem control signal is working
                                ;*      correctly.  It will only be performed if either 25 pin or staggered
                                ;*      loopback is specified.  This test uses the looped back signals CTS
                                ;*      and DCD to test the RTS signal.  This test is performed on all
                                ;*      active lines.
                                ;*
                                ;--*************************************************************************

031134                                  BGNTST
031134                                                                  T17::
                                ;+
                                ; Only perform this test if the DUT is in external or staggared loopback mode.
                                ;-
031134  032767  000002  151074          BIT     #BIT1,LOPBCK    ;CHECK TYPE OF LOOPBACK MODE SELECTED.
031142  001002                          BNE     1$
031144  000167  000456                  JMP     60$             ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
031150  000021                  1$:      TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031150  012767  000021  151110          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (79)
031156  012767  177777  151100          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
031164  012767  000001  152712          MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
031172  012767  017335  152706          MOV     #7901.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
031200  012767  007532  152702          MOV     #EM7901,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Reset the DUT to a known state, remove status codes from the fifo.
                                ;  Clear TX and  RX interrupt enable bits.
                                ; This subroutine reports error >>>>> 7901 <<<<<.
                                ;-
031206  004767  162274                  JSR     PC,CLNRST       ;RESET THE DUT.
031212  103402                          BCS     3$
031214  000167  000406                  JMP     60$             ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; Set up the TX/RX associated line number table.
                                ;-
031220  004767  161626          3$:     JSR     PC,ASLNTL       ;SET UP THE ASSOCIATED LINE TABLES.
                                ;+
                                ; Set up a loop which handles one line per iteration.
                                ; This loop clears all the RTSs and then sets them individually and waits for
                                ;   a response on the associated CTS and DCD signals.
                                ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                ;-
031224  005003                          CLR     R3              ;CLEAR THE LINE COUNTER.
031226  010300                  2$:     MOV     R3,R0
031230  00630C                          ASL     R0
031232  036067  002370  150774          BIT     BITTBL(R0),ACTLNS
031240  001465                          BEQ     8$              ;DON'T TEST IF NOT ACTIVE LINE.
                                ;+
                                ; Clear all the DUT LNCTRL registers RTS bits.
                                ;-
031242  005000                          CLR     R0              ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
031244  012705  000377                  MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
031250  004767  165436                  JSR     PC,WTWLNC       ;CLEAR ALL THE DUT RTS BITS.
031254  012704  000074                  MOV     #60.,R4
031260  004767  162434                  JSR     PC,DELAY        ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.
```

```
 7109                                   ;+
 7110                                   ; Check that at least one of associated DCD or CTS is clear and record states.
 7111                                   ;-
 7112 031264  116304  004044            MOVB    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
 7113 031270  010477  150746            MOV     R4,@CSRA        ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
 7114 031274  017705  150750            MOV     @STATA,R5       ;GET THE STATE OF THE ASSOCIATED DCD, CTS BITS.
 7115 031300  012700  014000            MOV     #BIT12!BIT11,R0
 7116 031304  040500                    BIC     R5,R0           ;CHECK FOR BOTH DCD AND CTS SET.
 7117 031306  001431                    BEQ     6$              ;GO REPORT RTS IS BAD IF BOTH ARE SET.
 7118                                   ;+
 7119                                   ; Set the RTS for the selected line and wait for either DCD or CTS to set.
 7120                                   ;-
 7121 031310  010377  150726            MOV     R3,@CSRA        ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
 7122 031314  052777  010000  150730    BIS     #BIT12,@LNCTRA  ;SET THE SELECTED LINE RTS.
 7123 031322  012701  130074            MOV     #130074,R1      ;SPECIFY TO WAIT UP TO 60 MS FOR CTS TO SET.
 7124 031326  032705  010000            BIT     #BIT12,R5       ;CHECK PREVIOUS STATE OF DCD BIT.
 7125 031332  001002                    BNE     4$              ;GO USE CTS IF DCD BIT WAS NOT CLEAR.
 7126 031334  012701  140074            MOV     #140074,R1      ;SPECIFY TO WAIT UP TO 60 MS FOR DCD SET.
 7127 031340  016702  150704    4$:     MOV     STATA,R2        ;SPECIFY TO LOOK IN STAT REG FOR BIT TO SET.
 7128 031344  010477  150672            MOV     R4,@CSRA        ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
 7129 031350  004767  165222            JSR     PC,WAIBIS       ;WAIT UP TO 60 MS FOR SIGNAL TO GO SET.
 7130 031354  103417                    BCS     8$              ;SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.
 7131 031356  017700  150666            MOV     @STATA,R0       ;GET THE STATUS REGISTER CONTENTS.
 7132 031362  042700  163777            BIC     #163777,R0      ;REMOVE ALL BUT THE DCD AND CTS BITS.
 7133 031366  040500                    BIC     R5,R0           ;TEST FOR SIGNAL ONCE CLEAR, BUT NOW SET.
 7134 031370  001011                    BNE     8$              ;GO LOOP IF SIGNAL HAS GONE FROM CLR TO SET.
 7135 031372                    6$:     ;Report RTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
 7136 031372  012767  017336  152506            MOV     #7902.,ERRNBR   ;SELECT THE ERROR NUMBER.
 7137 031400  012767  012040  152504            MOV     #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
 7138 031406  012701  007565            MOV     #EM7902,R1      ;SELECT THE ERROR MESSAGE.
 7139 031412                            ERROR                   ;                >>>>> ERROR <<<<<.
      031412  104460                                                                      TRAP    C$ERROR
 7140 031414  005203            8$:     INC     R3              ;SELECT THE NEXT LINE NUMBER.
 7141 031416  020327  000010            CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
 7142 031422  002701                    BLT     2$              ;LOOP IF NOT ALL LINES DONE.
 7143                                   ;+
 7144                                   ; Set up a loop which handles one line per iteration.
 7145                                   ; This loop sets all the RTSs and then clears them individually and waits for
 7146                                   ;   a response on the associated CTS and DCD signals.
 7147                                   ; This loop will clear the TX.IE and RX.IE bits if they are set.
 7148                                   ;-
 7149 031424  005003                    CLR     R3              ;CLEAR THE LINE COUNTER.
 7150 031426  010300            10$:    MOV     R3,R0
 7151 031430  006300                    ASL     R0
 7152 031432  036067  002370  150574    BIT     BITTBL(R0),ACTLNS
 7153 031440  001466                    BEQ     16$             ;DON'T TEST IF NOT ACTIVE LINE.
 7154                                   ;+
 7155                                   ; Set all the DUT LNCTRL registers RTS bits.
 7156                                   ;-
 7157 031442  012700  010000            MOV     #BIT12,R0       ;SPECIFY THAT RTS BITS ARE TO BE SET.
 7158 031446  012705  000377            MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
 7159 031452  004767  165234            JSR     PC,WTWLNC       ;SET ALL THE DUT RTS BITS.
 7160 031456  012704  000074            MOV     #60.,R4
 7161 031462  004767  162232            JSR     PC,DELAY        ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.
 7162                                   ;+
 7163                                   ; Check that at least one of associated DCD or CTS is set and record states.
 7164                                   ;-
```

```
 7165 031466  116304  004044                    MOVB    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
 7166 031472  010477  150544                    MOV     R4,@CSRA        ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
 7167 031476  017705  150546                    MOV     @STATA,R5       ;GET THE STATE OF THE ASSOCIATED DCD, CTS BITS.
 7168 031502  010500                            MOV     R5,R0
 7169 031504  042700  163777                    BIC     #163777,R0      ;CHECK FOR BOTH DCD AND CTS CLEAR.
 7170 031510  001431                           BEQ     14$             ;GO REPORT RTS IS BAD IF BOTH ARE CLEAR.
 7171                                 ;+
 7172                                 ; Clear the RTS for the selected line and wait for either DCD or CTS to clear.
 7173                                 ;-
 7174 031512  010377  150524                    MOV     R3,@CSRA        ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
 7175 031516  042777  010000  150526            BIC     #BIT12,@LNCTRA  ;CLEAR THE SELECTED LINE RTS.
 7176 031524  012701  130074                    MOV     #130074,R1      ;SPECIFY TO WAIT UP TO 60 MS FOR CTS TO CLEAR.
 7177 031530  032705  010000                    BIT     #BIT12,R5       ;CHECK PREVIOUS STATE OF DCD BIT.
 7178 031534  001402                           BEQ     12$             ;GO USE CTS IF DCD BIT WAS NOT SET.
 7179 031536  012701  140074                    MOV     #140074,R1      ;SPECIFY TO WAIT UP TO 60 MS FOR DCD CLEAR.
 7180 031542  016702  150502          12$:      MOV     STATA,R2        ;SPECIFY TO LOOK IN STAT REG FOR BIT TO CLR.
 7181 031546  010477  150470                    MOV     R4,@CSRA        ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
 7182 031552  004767  164744                    JSR     PC,WAIBIC       ;WAIT UP TO 60 MS FOR SIGNAL TO GO CLEAR.
 7183 031556  103417                           BCS     16$             ;SELECT NEXT LINE AND LOOP IF SIGNAL IS CLEAR.
 7184 031560  017700  150464                    MOV     @STATA,R0       ;GET THE STATUS REGISTER CONTENTS.
 7185 031564  042705  163777                    BIC     #163777,R5
 7186 031570  040005                            BIC     R0,R5           ;TEST FOR SIGNAL ONCE SET, BUT NOW CLEAR.
 7187 031572  001011                           BNE     16$             ;GO LOOP IF SIGNAL HAS GONE FROM SET TO CLR.
 7188 031574                          14$:    ;Report RTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
 7189 031574  012767  017337  152304            MOV     #7903.,ERRNBR   ;SELECT THE ERROR NUMBER.
 7190 031602  012767  012040  152302            MOV     #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
 7191 031610  012701  007565                    MOV     #EM7902,R1      ;SELECT THE ERROR MESSAGE.
 7192 031614                                    ERROR                   ;                >>>>> ERROR <<<<<.
      031614  104460                                                                            TRAP    C$ERROR
 7193 031616  005203                    16$:    INC     R3              ;SELECT THE NEXT LINE NUMBER.
 7194 031620  020327  000010                    CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
 7195 031624  002700                           BLT     10$             ;LOOP IF NOT ALL LINES DONE.
 7196
 7197 031626  005067  150432          60$:      CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7198
 7199 031632                                    ENDTST
      031632                                                                          L10043:
      031632  104401                                                                            TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - DSRMS -
                                ;+*****************************************************************************
                                ;*              - Data Set Ready Modem Signal Test -
                                ;*
                                ;*      This test verifies that the DSR modem status signal is working
                                ;*      correctly.  It will only be performed if either 25 pin or staggered
                                ;*      loopback is specified.  This test uses the looped back DTR signals
                                ;*      to test the DSR signal.  This test is performed on all the active
                                ;*      lines.
                                ;*
                                ;--*****************************************************************************

 031634                                 BGNTST
 031634                                                                  T18::
                                ;+
                                ; Only perform this test if the DUT is in external or staggared loopback mode.
                                ;-
 031634  032767  000002  150374          BIT     #BIT1,LOPBCK    ;CHECK TYPE OF LOOPBACK MODE SELECTED.
 031642  001002                         BNE     2$
 031644  000167  000372                 JMP     60$             ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
 031650  000022                 2$:     TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 031650  012767  000022  150410          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (80)
 031656  012767  177777  150400          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 031664  012767  000001  152212          MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
 031672  012767  017501  152206          MOV     #8001.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
 031700  012767  007616  152202          MOV     #EM8001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Reset the DUT to a known state, remove status codes from the FIFO.
                                ;  Clear TX and  RX interrupt enable bits.
                                ; This subroutine reports error >>>>> 8001 <<<<<.
                                ;-
 031706  004767  161574                 JSR     PC,CLNRST       ;RESET THE DUT.
 031712  103402                         BCS     4$
 031714  000167  000322                 JMP     60$             ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; Set up the TX/RX associated line number table.
                                ;-
 031720  004767  161126         4$:     JSR     PC,ASLNTL       ;SET UP THE ASSOCIATED LINE TABLES.
                                ;+
                                ; Set up a loop which handles one line per iteration.
                                ; This loop clears all the DTRs and then sets them individually and waits for
                                ;   a response on the associated DSR signal.
                                ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                ;-
 031724  005003                         CLR     R3              ;CLEAR THE LINE COUNTER.
 031726  010300                 6$:     MOV     R3,R0
 031730  006300                         ASL     R0
 031732  036067  002370  150274         BIT     BITTBL(R0),ACTLNS
 031740  001450                         BEQ     10$             ;DON'T TEST IF NOT ACTIVE LINE.
                                ;+
                                ; Clear all the DUT LNCTRL registers DTR bits.
                                ;-
 031742  005000                         CLR     R0              ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
 031744  012705  000377                 MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
 031750  004767  164736                 JSR     PC,WTWLNC       ;CLEAR ALL THE DUT DTR BITS.
 031754  012704  000050                 MOV     #40.,R4
 031760  004767  161734                 JSR     PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
```

```
                                        ;+
                                        ; Check that the specified DSR is clear.
                                        ;-
031764  010377  150252                    MOV    R3,@CSRA          ;SET IND.ADR.REG FIELD TO SELECTED LINE.
031770  032777  100000  150252            BIT    #BIT15,@STATA
031776  001020                           BNE    8$                ;GO REPORT DSR IS BAD IF BIT IS NOT CLEAR.
                                        ;+
                                        ; Set the DTR for the associated line.
                                        ; NOTE: If the associated line is not selected, DTR will not have been tested
                                        ;       in the DTR test (Only an issue in staggered loopback).
                                        ;-
032000  116304  004044                    MOVB   TXRLNB(R3),R4     ;GET THE ASSOCIATED LINE NUMBER.
032004  010477  150232                    MOV    R4,@CSRA          ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
032010  052777  001000  150234            BIS    #BIT9,@LNCTRA     ;SET THE ASSOCIATED LINE DTR.
                                        ;+
                                        ; Check that the selected line DSR is active.
                                        ;-
032016  010377  150220                    MOV    R3,@CSRA          ;SET IND.ADR.REG FIELD TO SELECTED LINE.
032022  012701  170050                    MOV    #170050,R1        ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
032026  016702  150216                    MOV    STATA,R2          ;PASS THE ADDRESS OF THE REGISTER TO TEST.
032032  004767  164540                   JSR    PC,WAIBIS         ;WAIT FOR DSR TO BECOME SET OR TIMEOUT.
032036  103411                           BCS    10$               ;SKIP ERROR REPORT IF SELECTED DSR IS SET.

032040                                  8$:    ;Report DSR MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
032040  012767  017502  152040            MOV    #8002.,ERRNBR     ;SELECT THE ERROR NUMBER.
032046  012767  012040  152036            MOV    #ER7801,ERRBLK    ;SELECT THE ERROR PRINT ROUTINE.
032054  012701  007654                    MOV    #EM8002,R1        ;SELECT THE ERROR MESSAGE.
032060                                   ERROR
032060  104460                                                     TRAP    C$ERROR
032062  005203                          10$:    INC    R3                ;SELECT THE NEXT LINE NUMBER.
032064  020327  000010                    CMP    R3,#NUMLNS        ;TEST FOR ALL LINES DONE.
032070  002716                           BLT    6$                ;LOOP IF NOT ALL LINES DONE.
                                        ;+
                                        ; Set up a loop which handles one line per iteration.
                                        ; This loop sets all the DTRs and then clears them individually and waits for
                                        ;   a response on the selected DSR signal.
                                        ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                        ;-
032072  005003                           CLR    R3                ;CLEAR THE LINE COUNTER.
032074  010300                          12$:    MOV    R3,R0
032076  006300                           ASL    R0
032100  036067  002370  150126           BIT    BITTBL(R0),ACTLNS
032106  001451                           BEQ    16$               ;DON'T TEST IF NOT ACTIVE LINE.
                                        ;+
                                        ; Set all the DUT LNCTRL registers DTR bits.
                                        ;-
032110  012700  001000                    MOV    #BIT9,R0          ;SPECIFY THAT DTR BITS ARE TO BE SET.
032114  012705  000377                    MOV    #MAPLNS,R5        ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
032120  004767  164566                   JSR    PC,WTWLNC         ;SET ALL THE DUT DTR BITS.
032124  012704  000050                    MOV    #40.,R4
032130  004767  161564                   JSR    PC,DELAY          ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
                                        ;+
                                        ; Check that the specified DSR is set.
                                        ;-
032134  010377  150102                    MOV    R3,@CSRA          ;SET IND.ADR.REG FIELD TO SELECTED LINE.
032140  032777  100000  150102            BIT    #BIT15,@STATA
```

```
 7313 032146  001420                        BEQ     14$             ;GO REPORT DSR IS BAD IF BIT IS NOT SET.
 7314                                ;+
 7315                                ; Clear the DTR for the associated line.
 7316                                ; NOTE: If the associated line is not selected, DTR will not have been tested
 7317                                ;       in the DTR test (Only an issue in staggered loopback).
 7318                                ;-
 7319 032150  116304  004044                 MOVB    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
 7320 032154  010477  150062                 MOV     R4,@CSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
 7321 032160  042777  001000  150064         BIC     #BIT9,@LNCTRA   ;CLEAR THE ASSOCIATED LINE DTR.
 7322                                ;+
 7323                                ; Check that the selected line DSR is clear.
 7324                                ;-
 7325 032166  010377  150050                 MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
 7326 032172  012701  170050                 MOV     #170050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
 7327 032176  016702  150046                 MOV     STATA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 7328 032202  004767  164314                JSR     PC,WAIBIC       ;WAIT FOR DSR TO BECOME CLEAR OR TIMEOUT.
 7329 032206  103411                        BCS     16$             ;SKIP ERROR REPORT IF SELECTED DSR IS CLEAR.
 7330
 7331 032210                        14$:    ;Report DSR MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
 7332 032210  012767  017503  151670         MOV     #8003.,ERRNBR   ;SELECT THE ERROR NUMBER.
 7333 032216  012767  012040  151666         MOV     #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
 7334 032224  012701  007654                 MOV     #EM8002,R1      ;SELECT THE ERROR MESSAGE.
 7335 032230                                ERROR
      032230  104460                                                        TRAP    C$ERROR
 7336 032232  005203                16$:     INC     R3              ;SELECT THE NEXT LINE NUMBER.
 7337 032234  020327  000010                 CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
 7338 032240  002715                        BLT     12$             ;LOOP IF NOT ALL LINES DONE.
 7339
 7340 032242  005067  150016         60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7341
 7342 032246                                ENDTST
      032246                                                            L10044:
      032246  104401                                                            TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RINGI -
                                ;++*************************************************************************
                                ;*              - Ring Indicator Modem Signal Test -
                                ;*
                                ;*      This test verifies that the RI modem status signal is working
                                ;*      correctly.  It will only be performed if either 25 pin or staggered
                                ;*      loopback is specified.  This test uses the looped back DTR signals
                                ;*      to test the RI signal.  This test is performed on all the active
                                ;*      lines.
                                ;*
                                ;--*************************************************************************

032250                                  BGNTST
032250                                                                  T19::
                                ;+
                                ; Only perform this test if the DUT is in external or staggared loopback mode.
                                ;-
032250  032767  000002  147760          BIT     #BIT1,LOPBCK    ;CHECK TYPE OF LOOPBACK MODE SELECTED.
032256  001002                        BNE     2$
032260  000167  000372                JMP     60$             ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
032264  000023                  2$:    TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
032264  012767  000023  147774        MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (81)
032272  012767  177777  147764        MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
032300  012767  000001  151576        MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
032306  012767  017645  151572        MOV     #8101.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
032314  012767  007720  151566        MOV     #EM8101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Reset the DUT to a known state, remove status codes from the FIFO.
                                ;  Clear TX and  RX interrupt enable bits.
                                ; This subroutine reports error >>>>> 8101 <<<<<.
                                ;-
032322  004767  161160                JSR     PC,CLNRST       ;RESET THE DUT.
032326  103402                        BCS     4$
032330  000167  000322                JMP     60$             ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; Set up the TX/RX associated line number table.
                                ;-
032334  004767  160512          4$:   JSR     PC,ASLNTL       ;SET UP THE ASSOCIATED LINE TABLES.
                                ;+
                                ; Set up a loop which handles one line per iteration.
                                ; This loop clears all the DTRs and then sets them individually and waits for
                                ;   a response on the associated RI signal.
                                ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                ;-
032340  005003                        CLR     R3              ;CLEAR THE LINE COUNTER.
032342  010300                  6$:   MOV     R3,R0
032344  006300                        ASL     R0
032346  036067  002370  147660        BIT     BITTBL(R0),ACTLNS
032354  001450                        BEQ     10$             ;DON'T TEST IF NOT ACTIVE LINE.
                                ;+
                                ; Clear all the DUT LNCTRL registers DTR bits.
                                ;-
032356  005000                        CLR     R0              ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
032360  012705  000377                MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
032364  004767  164322                JSR     PC,WTWLNC       ;CLEAR ALL THE DUT DTR BITS.
032370  012704  000050                MOV     #40.,R4
032374  004767  161320                JSR     PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
```

```
                                        ;+
                                        ; Check that the specified RI is clear.
                                        ;-
032400  010377  147636                          MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
032404  032777  020000  147636                   BIT     #BIT13,@STATA
032412  001020                                 BNE     8$              ;GO REPORT RI IS BAD IF BIT IS NOT CLEAR.
                                        ;+
                                        ; Set the DTR for the associated line.
                                        ; NOTE: If the associated line is not selected, DTR will not have been tested
                                        ;       in the DTR test (Only an issue in staggered loopback).
                                        ;-
032414  116304  004044                          MOVB    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
032420  010477  147616                          MOV     R4,@CSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
032424  052777  001000  147620                   BIS     #BIT9,@LNCTRA   ;SET THE ASSOCIATED LINE DTR.
                                        ;+
                                        ; Check that the selected line RI is active.
                                        ;-
032432  010377  147604                          MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
032436  012701  150050                          MOV     #150050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
032442  016702  147602                          MOV     STATA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
032446  004767  164124                        JSR     PC,WAIBIS       ;WAIT FOR RI TO BECOME SET OR TIMEOUT.
032452  103411                              BCS     10$             ;SKIP ERROR REPORT IF SELECTED RI IS SET.

032454                                  8$:     ;Report RI MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
032454  012767  017646  151424                    MOV     #8102.,ERRNBR   ;SELECT THE ERROR NUMBER.
032462  012767  012040  151422                    MOV     #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
032470  012701  007755                          MOV     #EM8102,R1      ;SELECT THE ERROR MESSAGE.
032474                                        ERROR
032474  104460                                                                  TRAP    C$ERROR
032476  005203                          10$:    INC     R3              ;SELECT THE NEXT LINE NUMBER.
032500  020327  000010                          CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
032504  002716                              BLT     6$              ;LOOP IF NOT ALL LINES DONE.
                                        ;+
                                        ; Set up a loop which handles one line per iteration.
                                        ; This loop sets all the DTRs and then clears them individually and waits for
                                        ;   a response on the selected RI signal.
                                        ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                        ;-
032506  005003                          CLR     R3              ;CLEAR THE LINE COUNTER.
032510  010300                          12$:    MOV     R3,R0
032512  006300                          ASL     R0
032514  036067  002370  147512                   BIT     BITTBL(R0),ACTLNS
032522  001451                              BEQ     16$             ;DON'T TEST IF NOT ACTIVE LINE.
                                        ;+
                                        ; Set all the DUT LNCTRL registers DTR bits.
                                        ;-
032524  012700  001000                          MOV     #BIT9,R0        ;SPECIFY THAT DTR BITS ARE TO BE SET.
032530  012705  000377                          MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
032534  004767  164152                        JSR     PC,WTWLNC       ;SET ALL THE DUT DTR BITS.
032540  012704  000050                          MOV     #40.,R4
032544  004767  161150                        JSR     PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
                                        ;+
                                        ; Check that the specified RI is set.
                                        ;-
032550  010377  147466                          MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
032554  032777  020000  147466                   BIT     #BIT13,@STATA
```

```
 7456 032562  001420                        BEQ     14$             ;GO REPORT RI IS BAD IF BIT IS NOT SET.
 7457                                ;+
 7458                                ; Clear the DTR for the associated line.
 7459                                ; NOTE: If the associated line is not selected, DTR will not have been tested
 7460                                ;       in the DTR test (Only an issue in staggered loopback).
 7461                                ;-
 7462 032564  116304  004044                 MOVB    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
 7463 032570  010477  147446                 MOV     R4,@CSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
 7464 032574  042777  001000  147450         BIC     #BIT9,@LNCTRA   ;CLEAR THE ASSOCIATED LINE DTR.
 7465                                ;+
 7466                                ; Check that the selected line RI is clear.
 7467                                ;-
 7468 032602  010377  147434                 MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
 7469 032606  012701  150050                 MOV     #150050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
 7470 032612  016702  147432                 MOV     STATA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 7471 032616  004767  163700                JSR     PC,WAIBIC       ;WAIT FOR RI TO BECOME CLEAR OR TIMEOUT.
 7472 032622  103411                        BCS     16$             ;SKIP ERROR REPORT IF SELECTED RI IS CLEAR.
 7473
 7474 032624                         14$:    ;Report RI MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
 7475 032624  012767  017647  151254         MOV     #8103.,ERRNBR   ;SELECT THE ERROR NUMBER.
 7476 032632  012767  012040  151252         MOV     #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
 7477 032640  012701  007755                 MOV     #EM8102,R1      ;SELECT THE ERROR MESSAGE.
 7478 032644                                ERROR
      032644  104460                                                     TRAP    C$ERROR
 7479 032646  005203                 16$:    INC     R3              ;SELECT THE NEXT LINE NUMBER.
 7480 032650  020327  000010                 CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
 7481 032654  002715                        BLT     12$             ;LOOP IF NOT ALL LINES DONE.
 7482
 7483 032656  005067  147402         60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7484
 7485 032662                                ENDTST
      032662                                                     L10045:
      032662  104401                                                    TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - CTSMS -
                                ;+*****************************************************************************
                                ;*              - Clear to Send Modem Signal Test -
                                ;*
                                ;*      This test verifies that the CTS modem status signal is working
                                ;*      correctly.  It will only be performed if either 25 pin or staggered
                                ;*      loopback is specified.  This test uses the looped back RTS signals
                                ;*      to test the CTS signal.  This test is performed on all the active
                                ;*      lines.
                                ;*
                                ;-*****************************************************************************

 7499 032664                            BGNTST
      032664                                                                    T20::
 7500                           ;+
 7501                           ; Only perform this test if the DUT is in external or staggared loopback mode.
 7502                           ;-
 7503 032664  032767  000002  147344     BIT     #BIT1,LOPBCK    ;CHECK TYPE OF LOOPBACK MODE SELECTED.
 7504 032672  001002             BNE     2$
 7505 032674  000167  000372     JMP     60$                     ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
 7506 032700  000024             2$:     TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 7507 032700  012767  000024  147360     MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (82)
 7508 032706  012767  177777  147350     MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 7509 032714  012767  000001  151162     MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
 7510 032722  012767  020011  151156     MOV     #8201.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
 7511 032730  012767  010020  151152     MOV     #EM8201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
 7512                           ;+
 7513                           ; Reset the DUT to a known state, remove status codes from the FIFO.
 7514                           ;  Clear TX and  RX interrupt enable bits.
 7515                           ; This subroutine reports error >>>>> 8201 <<<<<.
 7516                           ;-
 7517 032736  004767  160544             JSR     PC,CLNRST       ;RESET THE DUT.
 7518 032742  103402             BCS     4$
 7519 032744  000167  000322     JMP     60$                     ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
 7520                           ;+
 7521                           ; Set up the TX/RX associated line number table.
 7522                           ;-
 7523 032750  004767  160076     4$:     JSR     PC,ASLNTL       ;SET UP THE ASSOCIATED LINE TABLES.
 7524                           ;+
 7525                           ; Set up a loop which handles one line per iteration.
 7526                           ; This loop clears all the RTS's and then sets them individually and waits for
 7527                           ;   a response on the associated CTS signal.
 7528                           ; This loop will clear the TX.IE and RX.IE bits if they are set.
 7529                           ;-
 7530 032754  005003             CLR     R3              ;CLEAR THE LINE COUNTER.
 7531 032756  010300             6$:     MOV     R3,R0
 7532 032760  006300             ASL     R0
 7533 032762  036067  002370  147244     BIT     BITTBL(R0),ACTLNS
 7534 032770  001450             BEQ     10$             ;DON'T TEST IF NOT ACTIVE LINE.
 7535                           ;+
 7536                           ; Clear all the DUT LNCTRL registers RTS bits.
 7537                           ;-
 7538 032772  005000             CLR     R0              ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
 7539 032774  012705  000377     MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
 7540 033000  004767  163706     JSR     PC,WTWLNC       ;CLEAR ALL THE DUT RTS BITS.
 7541 033004  012704  000050     MOV     #40.,R4
 7542 033010  004767  160704     JSR     PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
```

```
                                        ;+
                                        ; Check that the specified CTS is clear.
                                        ;-
033014  010377  147222                          MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
033020  032777  004000  147222                  BIT     #BIT11,@STATA
033026  001020                                 BNE     8$              ;GO REPORT CTS IS BAD IF BIT IS NOT CLEAR.
                                        ;+
                                        ; Set the RTS for the associated line.
                                        ; NOTE: If the associated line is not selected, RTS will not have been tested
                                        ;       in the RTS test (Only an issue in staggered loopback).
                                        ;-
033030  116304  004044                          MOVB    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
033034  010477  147202                          MOV     R4,@CSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
033040  052777  010000  147204                  BIS     #BIT12,@LNCTRA  ;SET THE ASSOCIATED LINE RTS.
                                        ;+
                                        ; Check that the selected line CTS is active.
                                        ;-
033046  010377  147170                          MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
033052  012701  130050                          MOV     #130050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
033056  016702  147166                          MOV     STATA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
033062  004767  163510                         JSR     PC,WAIBIS       ;WAIT FOR CTS TO BECOME SET OR TIMEOUT.
033066  103411                                 BCS     10$             ;SKIP ERROR REPORT IF SELECTED CTS IS SET.

033070                                  8$:     ;Report CTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
033070  012767  020012  151010                    MOV     #8202.,ERRNBR   ;SELECT THE ERROR NUMBER.
033076  012767  012040  151006                    MOV     #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
033104  012701  010056                          MOV     #EM8202,R1      ;SELECT THE ERROR MESSAGE.
033110                                         ERROR
033110  104460                                                                  TRAP    C$ERROR
033112  005203                          10$:    INC     R3              ;SELECT THE NEXT LINE NUMBER.
033114  020327  000010                          CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
033120  002716                                 BLT     6$              ;LOOP IF NOT ALL LINES DONE.
                                        ;+
                                        ; Set up a loop which handles one line per iteration.
                                        ; This loop sets all the RTSs and then clears them individually and waits for
                                        ;   a response on the selected CTS signal.
                                        ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                        ;-
033122  005003                          12$:    CLR     R3              ;CLEAR THE LINE COUNTER.
033124  010300                                  MOV     R3,R0
033126  006300                                  ASL     R0
033130  036067  002370  147076                  BIT     BITTBL(R0),ACTLNS
033136  001451                                 BEQ     16$             ;DON'T TEST IF NOT ACTIVE LINE.
                                        ;+
                                        ; Set all the DUT LNCTRL registers RTS bits.
                                        ;-
033140  012700  010000                          MOV     #BIT12,R0       ;SPECIFY THAT RTS BITS ARE TO BE SET.
033144  012705  000377                          MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
033150  004767  163536                         JSR     PC,WTWLNC       ;SET ALL THE DUT RTS BITS.
033154  012704  000050                          MOV     #40.,R4
033160  004767  160534                         JSR     PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
                                        ;+
                                        ; Check that the specified CTS is set.
                                        ;-
033164  010377  147052                          MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
033170  032777  004000  147052                  BIT     #BIT11,@STATA
```

```
 7599 033176  001420                    BEQ     14$             ;GO REPORT CTS IS BAD IF BIT IS NOT SET.
 7600                                ;+
 7601                                ; Clear the RTS for the associated line.
 7602                                ; NOTE: If the associated line is not selected, RTS will not have been tested
 7603                                ;       in the RTS test (Only an issue in staggered loopback).
 7604                                ;-
 7605 033200  116304  004044         MOVB    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
 7606 033204  010477  147032         MOV     R4,@CSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
 7607 033210  042777  010000  147034 BIC     #BIT12,@LNCTRA  ;CLEAR THE ASSOCIATED LINE RTS.
 7608                                ;+
 7609                                ; Check that the selected line CTS is clear.
 7610                                ;-
 7611 033216  010377  147020         MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
 7612 033222  012701  130050         MOV     #130050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
 7613 033226  016702  147016         MOV     STATA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 7614 033232  004767  163264        JSR     PC,WAIBIC       ;WAIT FOR CTS TO BECOME CLEAR OR TIMEOUT.
 7615 033236  103411                BCS     16$             ;SKIP ERROR REPORT IF SELECTED CTS IS CLEAR.
 7616
 7617 033240                        14$:    ;Report CTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
 7618 033240  012767  020013  150640  MOV     #8203.,ERRNBR   ;SELECT THE ERROR NUMBER.
 7619 033246  012767  012040  150636  MOV     #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
 7620 033254  012701  010056         MOV     #EM8202,R1      ;SELECT THE ERROR MESSAGE.
 7621 033260                        ERROR
      033260  104460                                                     TRAP    C$ERROR
 7622 033262  005203                16$:    INC     R3              ;SELECT THE NEXT LINE NUMBER.
 7623 033264  020327  000010         CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
 7624 033270  002715                BLT     12$             ;LOOP IF NOT ALL LINES DONE.
 7625
 7626 033272  005067  146766        60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7627
 7628 033276                        ENDTST
      033276                                                     L10046:
      033276  104401                                                     TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - DCDMS -
                                ;+*****************************************************************************
                                ;*              - Data Carrier Detected Modem Signal Test -
                                ;*
                                ;*      This test verifies that the DCD modem status signal is working
                                ;*      correctly.  It will only be performed if either 25 pin or staggered
                                ;*      loopback is specified.  This test uses the looped back RTS signals
                                ;*      to test the DCD signal.  This test is performed on all the active
                                ;*      lines.
                                ;*
                                ;--*****************************************************************************

033300                                  BGNTST
033300                                                                  T21::
                                ;+
                                ; Only perform this test if the DUT is in external or staggered loopback mode.
                                ;-
033300  032767  000002  146730          BIT     #BIT1,LOPBCK    ;CHECK TYPE OF LOOPBACK MODE SELECTED.
033306  001002                        BNE     2$
033310  000167  000372                JMP     60$             ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
033314  000025                2$:     TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
033314  012767  000025  146744        MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (83)
033322  012767  177777  146734        MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
033330  012767  000001  150546        MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
033336  012767  020155  150542        MOV     #8301.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
033344  012767  010122  150536        MOV     #EM8301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Reset the DUT to a known state, remove status codes from the FIFO.
                                ;  Clear TX and  RX interrupt enable bits.
                                ; This subroutine reports error >>>>> 8301 <<<<<.
                                ;-
033352  004767  160130                JSR     PC,CLNRST       ;RESET THE DUT.
033356  103402                        BCS     4$
033360  000167  000322                JMP     60$             ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; Set up the TX/RX associated line number table.
                                ;-
033364  004767  157462        4$:     JSR     PC,ASLNTL       ;SET UP THE ASSOCIATED LINE TABLES.
                                ;+
                                ; Set up a loop which handles one line per iteration.
                                ; This loop clears all the RTSs and then sets them individually and waits for
                                ;   a response on the associated DCD signal.
                                ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                ;-
033370  005003                        CLR     R3              ;CLEAR THE LINE COUNTER.
033372  010300                6$:     MOV     R3,R0
033374  006300                        ASL     R0
033376  036067  002370  146630        BIT     BITTBL(R0),ACTLNS
033404  001450                        BEQ     10$             ;DON'T TEST IF NOT ACTIVE LINE.
                                ;+
                                ; Clear all the DUT LNCTRL registers RTS bits.
                                ;-
033406  005000                        CLR     R0              ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
033410  012705  000377                MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
033414  004767  163272                JSR     PC,WTWLNC       ;CLEAR ALL THE DUT RTS BITS.
033420  012704  000050                MOV     #40.,R4
033424  004767  160270                JSR     PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
```

```
                                        ;+
                                        ; Check that the specified DCD is clear.
                                        ;-
033430  010377  146606                          MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
033434  032777  010000  146606                  BIT     #BIT12,@STATA
033442  001020                                BNE     8$              ;GO REPORT DCD IS BAD IF BIT IS NOT CLEAR.
                                        ;+
                                        ; Set the RTS for the associated line.
                                        ; NOTE: If the associated line is not selected, RTS will not have been tested
                                        ;       in the RTS test (Only an issue in staggered loopback).
                                        ;-
033444  116304  004044                          MOVB    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
033450  010477  146566                          MOV     R4,@CSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
033454  052777  010000  146570                  BIS     #BIT12,@LNCTRA  ;SET THE ASSOCIATED LINE RTS.
                                        ;+
                                        ; Check that the selected line DCD is active.
                                        ;-
033462  010377  146554                          MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
033466  012701  140050                          MOV     #140050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
033472  016702  146552                          MOV     STATA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
033476  004767  163074                        JSR     PC,WAIBIS       ;WAIT FOR DCD TO BECOME SET OR TIMEOUT.
033502  103411                                BCS     10$             ;SKIP ERROR REPORT IF SELECTED DCD IS SET.

033504                                  8$:   ;Report DCD MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
033504  012767  020156  150374                  MOV     #8302.,ERRNBR   ;SELECT THE ERROR NUMBER.
033512  012767  012040  150372                  MOV     #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
033520  012701  010160                          MOV     #EM8302,R1      ;SELECT THE ERROR MESSAGE.
033524                                        ERROR
033524  104460                                                                  TRAP    C$ERROR
033526  005203                          10$:    INC     R3              ;SELECT THE NEXT LINE NUMBER.
033530  020327  000010                          CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
033534  002716                                BLT     6$              ;LOOP IF NOT ALL LINES DONE.
                                        ;+
                                        ; Set up a loop which handles one line per iteration.
                                        ; This loop sets all the RTSs and then clears them individually and waits for
                                        ;   a response on the selected DCD signal.
                                        ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                        ;-
033536  005003                          12$:    CLR     R3              ;CLEAR THE LINE COUNTER.
033540  010300                                  MOV     R3,R0
033542  006300                                  ASL     R0
033544  036067  002370  146462                  BIT     BITTBL(R0),ACTLNS
033552  001451                                BEQ     16$             ;DON'T TEST IF NOT ACTIVE LINE.
                                        ;+
                                        ; Set all the DUT LNCTRL registers RTS bits.
                                        ;-
033554  012700  010000                          MOV     #BIT12,R0       ;SPECIFY THAT RTS BITS ARE TO BE SET.
033560  012705  000377                          MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
033564  004767  163122                        JSR     PC,WTWLNC       ;SET ALL THE DUT RTS BITS.
033570  012704  000050                          MOV     #40.,R4
033574  004767  160120                        JSR     PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
                                        ;+
                                        ; Check that the specified DCD is set.
                                        ;-
033600  010377  146436                          MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
033604  032777  010000  146436                  BIT     #BIT12,@STATA
```

```
 7742 033612  001420                        BEQ     14$             ;GO REPORT DCD IS BAD IF BIT IS NOT SET.
 7743                                ;+
 7744                                ; Clear the RTS for the associated line.
 7745                                ; NOTE: If the associated line is not selected, RTS will not have been tested
 7746                                ;       in the RTS test (Only an issue in staggered loopback).
 7747                                ;-
 7748 033614  116304  004044                 MOVB    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
 7749 033620  010477  146416                 MOV     R4,@CSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
 7750 033624  042777  010000  146420         BIC     #BIT12,@LNCTRA  ;CLEAR THE ASSOCIATED LINE RTS.
 7751                                ;+
 7752                                ; Check that the selected line DCD is clear.
 7753                                ;-
 7754 033632  010377  146404                 MOV     R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
 7755 033636  012701  140050                 MOV     #140050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
 7756 033642  016702  146402                 MOV     STATA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 7757 033646  004767  162650                JSR     PC,WAIBIC       ;WAIT FOR DCD TO BECOME CLEAR OR TIMEOUT.
 7758 033652  103411                        BCS     16$             ;SKIP ERROR REPORT IF SELECTED DCD IS CLEAR.
 7759
 7760 033654                         14$:   ;Report DCD MODEM CONTROL SIGNAL DEFECTIVE ON LINE nn.
 7761 033654  012767  020157  150224         MOV     #8303.,ERRNBR   ;SELECT THE ERROR NUMBER.
 7762 033662  012767  012040  150222         MOV     #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
 7763 033670  012701  010160                 MOV     #EM8302,R1      ;SELECT THE ERROR MESSAGE.
 7764 033674                                ERROR
      033674  104460                                                                     TRAP    C$ERROR
 7765 033676  005203                 16$:    INC     R3              ;SELECT THE NEXT LINE NUMBER.
 7766 033700  020327  000010                 CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
 7767 033704  002715                        BLT     12$             ;LOOP IF NOT ALL LINES DONE.
 7768
 7769 033706  005067  146352         60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7770
 7771 033712                                ENDTST
      033712                                                                     L10047:
      033712  104401                                                                     TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - DTRINT -
                                ;+*****************************************************************************
                                ;*              - Data Terminal Ready Signal Interactions Test -
                                ;*
                                ;*      This test verifies that the DTR signal (and the looped back DSR and
                                ;*      RI status signals) do not interact with any other modem status signals.
                                ;*      It will only be performed if either 25 pin or staggered loopback is
                                ;*      specified.  This test is performed on all active lines.
                                ;*
                                ;-*****************************************************************************

033714                                  BGNTST
033714                                                                  T22::
                                ;+
                                ; Only perform this test if the DUT is in external or staggered loopback mode.
                                ;-
033714  032767  000002  146314          BIT     #BIT1,LOPBCK    ;CHECK TYPE OF LOOPBACK MODE SELECTED.
033722  001002                        BNE     2$
033724  000167  000352                JMP     60$             ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
033730  000026                  2$:     TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
033730  012767  000026  146330          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (84)
033736  012767  177777  146320          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
033744  012767  000001  150132          MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
033752  012767  020321  150126          MOV     #8401.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
033760  012767  010224  150122          MOV     #EM8401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Reset the DUT to a known state, remove status codes from the fifo.
                                ;  Clear TX and  RX interrupt enable bits.
                                ; This subroutine reports error >>>>> 8401 <<<<<.
                                ;-
033766  004767  157514                  JSR     PC,CLNRST       ;RESET THE DUT.
033772  103402                        BCS     4$
033774  000167  000302                JMP     60$             ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; Set up the TX/RX associated line number table.
                                ;-
034000  004767  157046          4$:     JSR     PC,ASLNTL       ;SET UP THE ASSOCIATED LINE TABLES.
                                ;+
                                ; Set up a loop which handles one line per iteration.
                                ; This loop clears all the DTRs and then sets them individually and checks
                                ;   for any responses on signals other than the associated RI and DSR signals.
                                ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                ;-
034004  005003                          CLR     R3              ;CLEAR THE LINE COUNTER.
034006  010300                  6$:     MOV     R3,R0
034010  006300                          ASL     R0
034012  036067  002370  146214          BIT     BITTBL(R0),ACTLNS
034020  001444                        BEQ     8$              ;DON'T TEST IF NOT ACTIVE LINE.
                                ;+
                                ; Clear all the DUT LNCTRL registers DTR bits.
                                ;-
034022  005000                          CLR     R0              ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
034024  012705  000377                  MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
034030  004767  162656                JSR     PC,WTWLNC       ;CLEAR ALL THE DUT DTR BITS.
034034  012704  000050                  MOV     #40.,R4
034040  004767  157654                JSR     PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
                                ;+
```

```
 7829                                   ; Record the states of the modem status signals.
 7830                                   ;-
 7831 034044  004767  161542                    JSR     PC,SAVMST       ;SAVE THE PRESENT MODEM STATUS STATES.
 7832                                   ;+
 7833                                   ; Set the DTR for the selected line.
 7834                                   ;-
 7835 034050  010377  146166                    MOV     R3,@CSRA        ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
 7836 034054  052777  001000  146170            BIS     #BIT9,@LNCTRA   ;SET THE SELECTED LINE DTR.
 7837 034062  012704  000050                    MOV     #40.,R4
 7838 034066  004767  157626                    JSR     PC,DELAY        ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
 7839                                   ;+
 7840                                   ; Check the present DUT STAT register contents against previous.
 7841                                   ; If any undesired changes have taken place, report the errors.
 7842                                   ;-
 7843 034072  116301  004044                    MOVB    TXRLNB(R3),R1   ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
 7844 034076  012702  120000                    MOV     #BIT15!BIT13,R2 ;IGNORE DSR AND RI ON ASSOCIATED LINE.
 7845 034102  004767  157444                    JSR     PC,CMPMST       ;COMPARE OLD AND NEW STAT CONTENTS.
 7846 034106  103411                            BCS     8$              ;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
 7847                                   ;Report INTERACTIONS FOUND BETWEEN DTR FOR LINE nn AND THE FOLLOWING SIGNALS:
 7848 034110  012767  020322  147770            MOV     #8402.,ERRNBR   ;SELECT THE ERROR NUMBER.
 7849 034116  012767  012066  147766            MOV     #ER8401,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
 7850 034124  012701  010300                    MOV     #EM8402,R1      ;SELECT THE DTR ERROR MESSAGES.
 7851 034130                                    ERROR                   ;ER8401 USES R1, R2, AND R3 VALUES.
      034130  104460                                                                    TRAP    C$ERROR
 7852                                   ;+
 7853                                   ; Select the next line and loop if not all possible lines have been handled.
 7854                                   ;-
 7855 034132  005203                    8$:     INC     R3              ;SELECT THE NEXT LINE NUMBER.
 7856 034134  020327  000010                    CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
 7857 034140  002722                            BLT     6$              ;LOOP IF NOT ALL LINES DONE.
 7858                                   ;+
 7859                                   ; Set up a loop which handles one line per iteration.
 7860                                   ; This loop sets all the DTRs and then clears them individually and checks
 7861                                   ;   for any responses on signals other than the associated RI and DSR signals.
 7862                                   ; This loop will clear the TX.IE and RX.IE bits if they are set.
 7863                                   ;-
 7864 034142  005003                            CLR     R3              ;CLEAR THE LINE COUNTER.
 7865 034144  010300                    10$:    MOV     R3,R0
 7866 034146  006300                            ASL     R0
 7867 034150  036067  002370  146056            BIT     BITTBL(R0),ACTLNS
 7868 034156  001445                            BEQ     12$             ;DON'T TEST IF NOT ACTIVE LINE.
 7869                                   ;+
 7870                                   ; Set all the DUT LNCTRL registers DTR bits.
 7871                                   ;-
 7872 034160  012700  001000                    MOV     #BIT9,R0        ;SPECIFY THAT DTR BITS ARE TO BE SET.
 7873 034164  012705  000377                    MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
 7874 034170  004767  162516                    JSR     PC,WTWLNC       ;SET ALL THE DUT DTR BITS.
 7875 034174  012704  000050                    MOV     #40.,R4
 7876 034200  004767  157514                    JSR     PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
 7877                                   ;+
 7878                                   ; Record the states of the modem status signals.
 7879                                   ;-
 7880 034204  004767  161402                    JSR     PC,SAVMST       ;SAVE THE PRESENT MODEM STATUS STATES.
 7881                                   ;+
 7882                                   ; Clear the DTR for the selected line.
 7883                                   ;-
 7884 034210  010377  146026                    MOV     R3,@CSRA        ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
```

```
 7885 034214  042777  001000  146030          BIC     #BIT9,@LNCTRA   ;CLEAR THE SELECTED LINE DTR.
 7886 034222  012704  000050                  MOV     #40.,R4
 7887 034226  004767  157466                  JSR     PC,DELAY        ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
 7888                                 ;+
 7889                                 ; Check the present DUT STAT register contents against previous.
 7890                                 ; If any undesired changes have taken place, report the errors.
 7891                                 ;-
 7892 034232  116301  004044                  MOVB    TXRLNB(R3),R1   ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
 7893 034236  012702  120000                  MOV     #BIT15!BIT13,R2 ;IGNORE DSR AND RI ON ASSOCIATED LINE.
 7894 034242  004767  157304                  JSR     PC,CMPMST       ;COMPARE OLD AND NEW STAT CONTENTS.
 7895 034246  103411                          BCS     12$             ;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
 7896                                 ;Report INTERACTIONS FOUND BETWEEN DTR FOR LINE nn AND THE FOLLOWING SIGNALS:
 7897 034250  012767  020323  147630          MOV     #8403.,ERRNBR   ;SELECT THE ERROR NUMBER.
 7898 034256  012767  012066  147626          MOV     #ER8401,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
 7899 034264  012701  010300                  MOV     #EM8402,R1      ;SELECT THE DTR ERROR MESSAGES.
 7900 034270                                  ERROR                   ;ER8401 USES R1, R2, AND R3 VALUES.
      034270  104460                                                          TRAP    C$ERROR
 7901                                 ;+
 7902                                 ; Select the next line and loop if not all possible lines have been handled.
 7903                                 ;-
 7904 034272  005203                  12$:    INC     R3              ;SELECT THE NEXT LINE NUMBER.
 7905 034274  020327  000010                  CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
 7906 034300  002721                          BLT     10$             ;LOOP IF NOT ALL LINES DONE.
 7907
 7908 034302  005067  145756          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7909
 7910 034306                                  ENDTST
      034306                                                                  L10050:
      034306  104401                                                                 TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RTSINT -
                                ;+*****************************************************************************
                                ;*              - Request To Send Signal Interactions Test -
                                ;*
                                ;*      This test verifies that the RTS signal (and the looped back DCD and CTS
                                ;*      status signals) do not interact with any other modem status signals.
                                ;*      It will only be performed if either 25 pin or staggered loopback is
                                ;*      specified.  This test is performed on all active lines.
                                ;*
                                ;--*****************************************************************************

034310                                  BGNTST
034310                                                                  T23::
                                ;+
                                ; Only perform this test if the DUT is in external or staggared loopback mode.
                                ;-
034310  032767  000002  145720          BIT     #BIT1,LOPBCK    ;CHECK TYPE OF LOOPBACK MODE SELECTED.
034316  001002                        BNE     2$
034320  000167  000352                JMP     60$             ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
034324  000027                  2$:     TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
034324  012767  000027  145734          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (85)
034332  012767  177777  145724          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
034340  012767  000001  147536          MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
034346  012767  020465  147532          MOV     #8501.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
034354  012767  010323  147526          MOV     #EM8501,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Reset the DUT to a known state, remove status codes from the fifo.
                                ;  Clear TX and  RX interrupt enable bits.
                                ; This subroutine reports error >>>>> 8501 <<<<<.
                                ;-
034362  004767  157120                  JSR     PC,CLNRST       ;RESET THE DUT.
034366  103402                        BCS     4$
034370  000167  000302                JMP     60$             ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; Set up the TX/RX associated line number table.
                                ;-
034374  004767  156452          4$:     JSR     PC,ASLNTL       ;SET UP THE ASSOCIATED LINE TABLES.
                                ;+
                                ; Set up a loop which handles one line per iteration.
                                ; This loop clears all the RTSs and then sets them individually and checks
                                ;   for any responses on signals other than the associated DCD and CTS signals.
                                ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                ;-
034400  005003                          CLR     R3              ;CLEAR THE LINE COUNTER.
034402  010300                  6$:     MOV     R3,R0
034404  006300                          ASL     R0
034406  036067  002370  145620          BIT     BITTBL(R0),ACTLNS
034414  001444                        BEQ     8$              ;DON'T TEST IF NOT ACTIVE LINE.
                                ;+
                                ; Clear all the DUT LNCTRL registers RTS bits.
                                ;-
034416  005000                          CLR     R0              ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
034420  012705  000377                  MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
034424  004767  162262                JSR     PC,WTWLNC       ;CLEAR ALL THE DUT RTS BITS.
034430  012704  000050                  MOV     #40.,R4
034434  004767  157260                JSR     PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
                                ;+
```

```
                                        ; Record the states of the modem status signals.
                                        ;-
034440  004767  161146                          JSR     PC,SAVMST       ;SAVE THE PRESENT MODEM STATUS STATES.
                                        ;+
                                        ; Set the RTS for the selected line.
                                        ;-
034444  010377  145572                          MOV     R3,@CSRA        ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
034450  052777  010000  145574                  BIS     #BIT12,@LNCTRA  ;SET THE SELECTED LINE RTS.
034456  012704  000050                          MOV     #40.,R4
034462  004767  157232                        JSR     PC,DELAY        ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
                                        ;+
                                        ; Check the present DUT STAT register contents against previous.
                                        ; If any undesired changes have taken place, report the errors.
                                        ;-
034466  116301  004044                          MOVB    TXRLNB(R3),R1   ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
034472  012702  014000                          MOV     #BIT12!BIT11,R2 ;IGNORE DCD AND CTS ON ASSOCIATED LINE.
034476  004767  157050                        JSR     PC,CMPMST       ;COMPARE OLD AND NEW STAT CONTENTS.
034502  103411                                BCS     8$              ;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
                                        ;Report INTERACTIONS FOUND BETWEEN RTS FOR LINE nn AND THE FOLLOWING SIGNALS:
034504  012767  020466  147374                  MOV     #8502.,ERRNBR   ;SELECT THE ERROR NUMBER.
034512  012767  012066  147372                  MOV     #ER8401,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
034520  012701  010377                          MOV     #EM8502,R1      ;SELECT THE RTS ERROR MESSAGES.
034524                                        ERROR                   ;ER1901 USES R1, R2, AND R3 VALUES.
034524  104460                                                                  TRAP    C$ERROR
                                        ;+
                                        ; Select the next line and loop if not all possible lines have been handled.
                                        ;-
034526  005203                          8$:     INC     R3              ;SELECT THE NEXT LINE NUMBER.
034530  020327  000010                          CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
034534  002722                                BLT     6$              ;LOOP IF NOT ALL LINES DONE.
                                        ;+
                                        ; Set up a loop which handles one line per iteration.
                                        ; This loop sets all the RTSs and then clears them individually and checks
                                        ;   for any responses on signals other than the associated DCD and CTS signals.
                                        ; This loop will clear the TX.IE and RX.IE bits if they are set.
                                        ;-
034536  005003                                  CLR     R3              ;CLEAR THE LINE COUNTER.
034540  010300                          10$:    MOV     R3,R0
034542  006300                                  ASL     R0
034544  036067  002370  145462                  BIT     BITTBL(R0),ACTLNS
034552  001445                                BEQ     12$             ;DON'T TEST IF NOT ACTIVE LINE.
                                        ;+
                                        ; Set all the DUT LNCTRL registers RTS bits.
                                        ;-
034554  012700  010000                          MOV     #BIT12,R0       ;SPECIFY THAT RTS BITS ARE TO BE SET.
034560  012705  000377                          MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
034564  004767  162122                        JSR     PC,WTWLNC       ;SET ALL THE DUT RTS BITS.
034570  012704  000050                          MOV     #40.,R4
034574  004767  157120                        JSR     PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
                                        ;+
                                        ; Record the states of the modem status signals.
                                        ;-
034600  004767  161006                          JSR     PC,SAVMST       ;SAVE THE PRESENT MODEM STATUS STATES.
                                        ;+
                                        ; Clear the RTS for the selected line.
                                        ;-
034604  010377  145432                          MOV     R3,@CSRA        ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
```

HARDWARE TEST - RTSINT -

```
 8024 034610  042777  010000  145434          BIC     #BIT12,@LNCTRA  ;CLEAR THE SELECTED LINE RTS.
 8025 034616  012704  000050                  MOV     #40.,R4
 8026 034622  004767  157072                  JSR     PC,DELAY        ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
 8027                                ;+
 8028                                ; Check the present DUT STAT register contents against previous.
 8029                                ; If any undesired changes have taken place, report the errors.
 8030                                ;-
 8031 034626  116301  004044                  MOVB    TXRLNB(R3),R1   ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
 8032 034632  012702  014000                  MOV     #BIT12!BIT11,R2 ;IGNORE DCD AND CTS ON ASSOCIATED LINE.
 8033 034636  004767  156710                  JSR     PC,CMPMST       ;COMPARE OLD AND NEW STAT CONTENTS.
 8034 034642  103411                          BCS     12$             ;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
 8035                                ;Report INTERACTIONS FOUND BETWEEN RTS FOR LINE nn AND THE FOLLOWING SIGNALS:
 8036 034644  012767  020467  147234          MOV     #8503.,ERRNBR   ;SELECT THE ERROR NUMBER.
 8037 034652  012767  012066  147232          MOV     #ER8401,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
 8038 034660  012701  010377                  MOV     #EM8502,R1      ;SELECT THE RTS ERROR MESSAGES.
 8039 034664                                  ERROR                   ;ER1901 USES R1, R2, AND R3 VALUES.
      034664  104460                                                          TRAP    C$ERROR
 8040                                ;+
 8041                                ; Select the next line and loop if not all possible lines have been handled.
 8042                                ;-
 8043 034666  005203                  12$:    INC     R3              ;SELECT THE NEXT LINE NUMBER.
 8044 034670  020327  000010                  CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
 8045 034674  002721                          BLT     10$             ;LOOP IF NOT ALL LINES DONE.
 8046
 8047 034676  005067  145362          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 8048
 8049 034702                                  ENDTST
      034702                                                      L10051:
      034702  104401                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - TXLNS -
                                ;+*****************************************************************************
                                ;*                        - TX Lines Test -
                                ;*
                                ;*      This test verifies that the TX lines and RX lines are working correctly
                                ;*      through the device cables, distribution panel (if installed), and
                                ;*      loopback connector(s).  It will only be performed if either 25 pin or
                                ;*      staggered loopback is specified.  This test sends a character on each
                                ;*      active line and verifies that the proper character is detected on each
                                ;*      receive line.
                                ;*
                                ;--*****************************************************************************

034704                                  BGNTST
034704                                                                  T24::
                                ;+
                                ; Only perform this test if the DUT is in external or staggared loopback mode.
                                ;-
034704  032767  000002  145324          BIT     #BIT1,LOPBCK    ;CHECK TYPE OF LOOPBACK MODE SELECTED.
034712  001002                        BNE     2$
034714  000167  000274                JMP     60$             ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
034720  000030                2$:     TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
034720  012767  000030  145340          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (86)
034726  012767  177777  145330          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
034734  012767  000001  147142          MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
034742  012767  020631  147136          MOV     #8601.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
034750  012767  010403  147132          MOV     #EM8601,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Reset the DUT to a known state, remove status codes from the fifo.
                                ;  Clear TX and  RX interrupt enable bits.
                                ; This subroutine reports error >>>>> 8601 <<<<<.
                                ;-
034756  004767  156524                JSR     PC,CLNRST       ;RESET THE DUT.
034762  103402                        BCS     4$
034764  000167  000224                JMP     60$             ;ABORT  THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; Set up the TX/RX associated line number table.
                                ;-
034770  004767  156056          4$:     JSR     PC,ASLNTL       ;SET UP THE ASSOCIATED LINE TABLES.
                                ;+
                                ; Set up transmission an reception parameters for all lines.
                                ; 9600 baud,8 char,1 stopbit,no parity.
                                ;-
034774  012705  000377                  MOV     #MAPLNS,R5      ;INDICATE ALL LINES TO BE SET UP.
035000  005000                        CLR     R0              ;SPECIFY ALL LNCTRL BITS TO BE CLEARED.
035002  004767  161704                JSR     PC,WTWLNC       ;CLEAR ALL LNCTRL BITS.
035006  012700  156430                MOV     #156430,R0      ;SET UP BAUD RATE,ETC.
035012  004767  161724                JSR     PC,WTWLPR       ;SET COMMUNICATION PARAMETERS ON ALL LINES.
                                ;+
                                ; Enable transmitters on all active lines.
                                ;-
035016  016705  145212                  MOV     ACTLNS,R5       ;PASS ACTIVE LINE BIT MAP.
035022  004767  161204                JSR     PC,TXENBL       ;ENABLE TRANSMISSIONS ON ALL LINES.
                                ;+
                                ; Enable reception on all lines associated with active lines.
                                ;-
035026  016705  145202                  MOV     ACTLNS,R5       ;GET ACTIVE LINES BIT MAP TO CONVERT.
```

```
 8107 035032  004767  156606                JSR     PC,CONMAP       ;CONVERT TO ASSOCIATED LINE MAP.
 8108 035036  012700  000004                 MOV     #BIT2,R0        ;INDICATE RX.ENABLE BIT TO BE SET.
 8109 035042  004767  161644                JSR     PC,WTWLNC       ;ENABLE RX ON THE ASSOCIATED LINES.
 8110                                ;+
 8111                                ; Set up a loop which sends a character on each active line.
 8112                                ;-
 8113 035046  005003                        CLR     R3              ;CLEAR THE LINE COUNTER.
 8114 035050  010300                6$:     MOV     R3,R0
 8115 035052  006300                        ASL     R0
 8116 035054  036067  002370  145152        BIT     BITTBL(R0),ACTLNS
 8117 035062  001406                       BEQ     8$              ;DON'T SEND IF NOT ACTIVE LINE.
 8118 035064  010377  145152                MOV     R3,@CSRA        ;SET UP THE IND.ADR.REG FIELD FOR PROPER LINE.
 8119 035070  052700  100000                BIS     #BIT15,R0       ;OR IN THE TX.DATA.VALID BIT TO THE TX CHAR.
 8120 035074  010077  145144                MOV     R0,@TXCHA       ;SEND THE CHARACTER.
 8121 035100  005203                8$:     INC     R3              ;SELECT THE NEXT LINE NUMBER.
 8122 035102  020327  000010                CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
 8123 035106  002760                       BLT     6$              ;LOOP IF NOT ALL LINES DONE.
 8124                                ;+
 8125                                ; Set up loop which waits for all chars to be received or time-out to occur.
 8126                                ;-
 8127 035110  016705  145120                MOV     ACTLNS,R5
 8128 035114  012701  000062                MOV     #50.,R1         ;SELECT 50 MS TIME-OUT VALUE.
 8129 035120  012702  100000                MOV     #BIT15,R2       ;SELECT DATA.VALID BIT TO BE TESTED.
 8130 035124  016704  145114                MOV     RBUFA,R4        ;SPECIFY THAT BIT IS FOUND IN RBUF REG.
 8131 035130  010203                10$:    MOV     R2,R3           ;SPECIFY TO WAIT FOR BIT TO BE SET.
 8132 035132  004767  157236                JSR     PC,MSLGET       ;WAIT FOR A VALID CHAR TO BE RECEIVED.
 8133 035136  103016                       BCC     12$             ;GO ANALYSE PROBLEM IF TIME-OUT.
 8134 035140  010003                        MOV     R0,R3           ;TEST THE RX CHARACTER TO VERIFY THAT
 8135 035142  000303                        SWAB    R3              ;  IT IS A VALID COMBINATION OF LINE
 8136 035144  042700  177600                BIC     #177600,R0      ;  NUMBER AND DATA.  IGNORE ANY ERROR
 8137 035150  042703  177760                BIC     #177760,R3      ;  BITS WHICH ARE SET.
 8138 035154  006303                        ASL     R3              ;
 8139 035156  026300  004004                CMP     TXRXLB(R3),R0   ;
 8140 035162  001362                       BNE     10$             ;RX CHAR A VALID COMBINATION?  NO, IGNORE IT.
 8141 035164  046005  002370                BIC     BITTBL(R0),R5   ;YES, CLEAR THE BIT FOR THE ASSOCIATED LINE.
 8142 035170  001357                       BNE     10$             ;ALL ACTIVE LINES DONE?  NO, LOOP.
 8143 035172  000410                        BR      60$             ;YES, SUCCESSFUL COMPLETION.  EXIT THE TEST.
 8144
 8145                                ;+
 8146                                ; Not all active lines received correct characters.
 8147                                ; Report "TX/RX LINE ERROR ON THE FOLLOWING LOOPED BACK TX LINES:"
 8148                                ;   Error number                >>>>> 8602 <<<<<.
 8149                                ;-
 8150 035174  005267  146706        12$:    INC     ERRNBR          ;SELECT PROPER ERROR NUMBER (8602).
 8151 035200  012767  011742  146704        MOV     #ER6401,ERRBLK  ;SELECT PROPER ERROR REPORTING ROUTINE.
 8152 035206  012701  010464                MOV     #EM8602,R1      ;SELECT PROPER ERROR MESSAGE.
 8153 035212                               ERROR
      035212  104460                                                                    TRAP    C$ERROR
 8154
 8155 035214  005067  145044        60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 8156
 8157 035220                               ENDTST
      035220                                                                    L10052:
      035220  104401                                                                    TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - TXLINT -
                                ;+*****************************************************************************
                                ;*                      - TX Lines Interactions Test -
                                ;*
                                ;*      This test verifies that each TX line does not interact with any other
                                ;*      TX lines or modem control signals.  It will only be performed if
                                ;*      either 25 pin or staggered loopback is specified.  This test causes
                                ;*      a BREAK condition on each active line individually and looks for any
                                ;*      characters generated on other lines or any changes to modem control
                                ;*      signals on active lines.
                                ;*
                                ;--*****************************************************************************

035222                                  BGNTST
035222                                                                  T25::
                                ;+
                                ; Only perform this test if the DUT is in external or staggared loopback mode.
                                ;-
035222  032767  000002  145006          BIT     #BIT1,LOPBCK    ;CHECK TYPE OF LOOPBACK MODE SELECTED.
035230  001002                        BNE     2$
035232  000167  000376                JMP     60$             ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
035236  000031                  2$:    TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
035236  012767  000031  145022          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (87)
035244  012767  177777  145012          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
035252  012767  000001  146624          MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
035260  012767  020775  146620          MOV     #8701.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
035266  012767  010554  146614          MOV     #EM8701,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; Reset the DUT to a known state, remove status codes from the fifo.
                                ;  Clear TX and  RX interrupt enable bits.
                                ; This subroutine reports error >>>>> 8701 <<<<<.
                                ;-
035274  004767  156206                  JSR     PC,CLNRST       ;RESET THE DUT.
035300  103402                        BCS     4$
035302  000167  000326                JMP     60$             ;ABORT  THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; Set up the TX/RX associated line number table.
                                ;-
035306  004767  155540          4$:     JSR     PC,ASLNTL       ;SET UP THE ASSOCIATED LINE TABLES.
                                ;+
                                ; Set up transmission an reception parameters for all lines.
                                ; 9600 baud,8 char,1 stopbit,no parity.
                                ;-
035312  012705  000377                  MOV     #MAPLNS,R5      ;INDICATE ALL LINES TO BE SET UP.
035316  005000                        CLR     R0              ;SPECIFY ALL LNCTRL BITS TO BE CLEARED.
035320  004767  161366                JSR     PC,WTWLNC       ;CLEAR ALL LNCTRL BITS.
035324  012700  156430                MOV     #156430,R0      ;SET UP BAUD RATE,ETC.
035330  004767  161406                JSR     PC,WTWLPR       ;SET COMMUNICATION PARAMETERS ON ALL LINES.
                                ;+
                                ; Enable transmitters on all active lines.
                                ;-
035334  016705  144674                  MOV     ACTLNS,R5       ;PASS ACTIVE LINE BIT MAP.
035340  004767  160666                JSR     PC,TXENBL       ;ENABLE TRANSMISSIONS ON ALL LINES.
                                ;+
                                ; Enable reception on all lines associated with active lines.
                                ;-
035344  016705  144664                  MOV     ACTLNS,R5       ;GET THE ACTIVE LINE MAP TO CONVERT.
```

```
 8215 035350  004767  156270                JSR     PC,CONMAP       ;CONVERT TO ASSOCIATED LINE MAP.
 8216 035354  012700  000004                 MOV     #BIT2,R0        ;INDICATE RX.ENABLE BIT TO BE SET.
 8217 035360  004767  161326                JSR     PC,WTWLNC       ;ENABLE RX ON THE ASSOCIATED LINES.
 8218                                ;+
 8219                                ; Set up a loop which sets the BREAK bit on each active line individually and
 8220                                ;    checks for interactions.
 8221                                ;-
 8222 035364  005003                         CLR     R3              ;CLEAR THE LINE COUNTER.
 8223 035366  010300                 6$:     MOV     R3,R0
 8224 035370  006300                         ASL     R0
 8225 035372  036067  002370  144634         BIT     BITTBL(R0),ACTLNS
 8226 035400  001503                        BEQ     16$             ;LINE ACTIVE?  NO, SKIP THE LINE.
 8227                                    ;+
 8228                                    ; Clear the BREAK bits for all lines.
 8229                                    ; Delay 10 ms to all lines to get out of BREAK condition.
 8230                                    ;-
 8231 035402  005000                         CLR     R0              ;CLEAR LINE COUNTER.
 8232 035404  010077  144632         8$:     MOV     R0,@CSRA        ;SET UP THE IND.ADR.REG FILED FOR PROPER LINE.
 8233 035410  042777  000010  144634         BIC     #BIT3,@LNCTRA   ;CLEAR A BREAK BIT.
 8234 035416  005200                         INC     R0              ;SET LINE COUNTER TO NEXT LINE.
 8235 035420  020027  000010                 CMP     R0,#NUMLNS
 8236 035424  001367                        BNE     8$              ;DONE?  NO, LOOP TO DO NEXT LINE.
 8237 035426  012704  000012                 MOV     #10.,R4         ;SELECT 10 MS DELAY.
 8238 035432  004767  156262                JSR     PC,DELAY        ;DELAY TO ALLOW BREAKS TO BE CLEARED.
 8239                                    ;+
 8240                                    ; Record the states of the modem control signals.
 8241                                    ; Purge the FIFO of any unwanted characters.
 8242                                    ; This routine reports errors with numbers >>>>> 8702 thru 8704 <<<<<.
 8243                                    ;-
 8244 035436  004767  160150                JSR     PC,SAVMST       ;RECORD THE STATES OF MODEM STATUS SIGNALS.
 8245 035442  012767  020776  146436         MOV     #8702.,ERRNBR
 8246 035450  004767  157442                JSR     PC,PUFIFR       ;PURGE THE FIFO.
 8247 035454  103067                        BCC     60$             ;PURGE SUCCESSFUL?  NO, EXIT TEST.
 8248                                    ;+
 8249                                    ; Set the break on the selected line.
 8250                                    ;-
 8251 035456  010377  144560                MOV     R3,@CSRA        ;SET UP THE IND.ADR.REG FIELD FOR PROPER LINE.
 8252 035462  052777  000010  144562         BIS     #BIT3,@LNCTRA   ;SET THE BREAK BIT IN LNCTRL REGISTER.
 8253 035470  012704  000C12                 MOV     #10.,R4         ;SELECT 10 MS DELAY.
 8254 035474  004767  156220                JSR     PC,DELAY        ;DELAY TO ALLOW BREAK TO BE SET.
 8255                                    ;+
 8256                                    ; Verify that only the associated line receives characters.
 8257                                    ;-
 8258 035500  012700  000013                 MOV     #11.,R0         ;ALLOW UP TO 10 "LEGAL" CHARACTERS.
 8259 035504  017701  144534         10$:    MOV     @RBUFA,R1       ;READ A CHARACTER FROM THE DUT.
 8260 035510  100022                        BPL     14$             ;DATA VALID?  NO, CHECK MODEM SIGNALS.
 8261 035512  000301                         SWAB    R1
 8262 035514  042701  177760                BIC     #177760,R1      ;GET LINE NUMBER OF RX CHAR.
 8263 035520  126103  004044                CMPB    TXRLNB(R1),R3   ;COMPARE AGAINST TX ASSOCIATED LINE NUMBER.
 8264 035524  001003                        BNE     12$             ;IS CHARACTER ON ASSOCIATED LINE?  NO, ERROR.
 8265 035526  005300                         DEC     R0              ;COUNT THIS CHARACTER.
 8266 035530  001365                        BNE     10$             ;TOO MANY RXED?  NO, LOOP TO LOOK FOR MORE.
 8267 035532  000433                        BR      50$             ;YES, REPORT ERROR AND ABORT.
 8268                                    ;+
 8269                                    ; Character received on wrong line.
 8270                                    ; Report "DATA LINE INTERACTIONS ON LINE nn DECIMAL."
 8271                                    ;   Error number            >>>>> 8705 <<<<<.
```

HARDWARE TEST - TXLINT -

```
8272                                            ;-
8273 035534  012767  021001  146344  12$:      MOV     #8705.,ERRNBR
8274 035542  012701  010614                    MOV     #EM8702,R1      ;SELECT THE ERROR MESSAGE.
8275 035546  012767  012040  146336            MOV     #ER7801,ERRBLK  ;SELECT THE ERROR REPORT ROUTINE.
8276 035554                                    ERROR
     035554  104460                                                    TRAP    C$ERROR
8277                                            ;+
8278                                            ; Check for modem signal interactions.
8279                                            ;-
8280 035556  005002                    14$:      CLR     R2              ;SELECT NO BITS TO BE IGNORED.
8281 035560  004767  155766                    JSR     PC,CMPMST       ;COMPARE OLD AND NEW MODEM STATUS.
8282 035564  103411                            BCS     16$             ;STATUS CHANGES?  NO, SKIP ERROR REPORT.
8283                                            ;+
8284                                            ; Modem status changes have been discovered.
8285                                            ; Report                >>>>> 8706 <<<<<.
8286                                            ;  "INTERACTIONS FOUND BETWEEN TX FOR LINE nn AND THE FOLLOWING SIGNALS:"
8287                                            ;-
8288 035566  012767  021002  146312            MOV     #8706.,ERRNBR
8289 035574  012767  012066  146310            MOV     #ER8401,ERRBLK
8290 035602  012701  010643                    MOV     #EM8703,R1
8291 035606                                    ERROR                   ;ERR8401 USES R1, R2, R3 VALUES.
     035606  104460                                                    TRAP    C$ERROR
8292                                            ;+
8293                                            ; Select next line and test it if we are not done.
8294                                            ;-
8295 035610  005203                    16$:      INC     R3
8296 035612  020327  000010                    CMP     R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
8297 035616  002663                            BLT     6$              ;ALL DONE?  NO, LOOP TO DO NEXT LINE.
8298 035620  000405                            BR      60$             ;YES, EXIT THE TEST.
8299                                            ;+
8300                                            ; Found an error not directly related to the test.
8301                                            ; Report error                  >>>>> 8707 <<<<<.
8302                                            ; Abort the test.
8303                                            ;-
8304 035622  012767  021003  146256  50$:      MOV     #8707.,ERRNBR
8305 035630  004767  160146                    JSR     PC,TSABRT
8306
8307 035634  005067  144424            60$:      CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
8308
8309 035640                                    ENDTST
     035640                                                                    L10053:
     035640  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.
                                ;*
                                ;-- ******************************************************************************
035642                                  BGNTST
035642                                                                          T26::
000032                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
035642  012767  000032  144416   MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (93)
035650  012767  177777  144406   MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
035656  016702  144560           MOV    BMPCQP,R2       ;GET THE CONTENTS OF THE POINTER.
035662  012703  002444           MOV    #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
035666  020203                   CMP    R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
035670  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"

035672  012701  011244          MOV     #EM9304,R1      ;PASS THE FIRST MESSAGE TO BE REORTED.
035676                          ERRDF   9301,EM9301,ER9301 ;           >>>>> ERROR #9301 <<<<<.
035676  104455                                                                  TRAP    C$ERDF
035700  022125                                                                  .WORD   9301
035702  011127                                                                  .WORD   EM9301
035704  012620                                                                  .WORD   ER9301

035706  012767  002444  144526           MOV     #BMPCQB,BMPCQP  ;SET POINTER BACK TO THE BEGINING OF THE QUE.

035714  005067  144344          60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
035720                          ENDTST
035720                                                                  L10054:
035720  104401                                                                  TRAP    C$ETST
```

```
                        ;*****************************************************************************
                        ;
                        ;               VDHB.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.
                        ;--

 8364 035722                    BGNHRD
      035722  000027                                                .WORD L10055-L$HARD/2
      035724                                                  L$HARD::
 8365
 8375                   ;DEVICE CSR ADDRESS QUESTION:
 8376 035724                    GPRMA   HWPTQ1,0,0,160000,177776,YES
      035724  000031                                            .WORD   T$CODE
      035726  036002                                            .WORD   HWPTQ1
      035730  160000                                            .WORD   T$LOLIM
      035732  177776                                            .WORD   T$HILIM
 8377                   ;DEVICE INTERRUPT VECTOR QUESTION:
 8378 035734                    GPRMA   HWPTQ2,2,0,40,776,YES
      035734  001031                                            .WORD   T$CODE
      035736  036020                                            .WORD   HWPTQ2
      035740  000040                                            .WORD   T$LOLIM
      035742  000776                                            .WORD   T$HILIM
 8379                   ;ACTIVE LINES BIT MAP QUESTION:
 8380 035744                    GPRMD   HWPTQ3,4,0,MAPLNS,0,177777,YES
      035744  002032                                            .WORD   T$CODE
      035746  036053                                            .WORD   HWPTQ3
      035750  000377                                            .WORD   MAPLNS
      035752  000000                                            .WORD   T$LOLIM
      035754  177777                                            .WORD   T$HILIM
 8381                   ;TYPE OF LOOPBACK QUESTION:
 8382 035756                    GPRMD   HWPTQ4,6,0,377,1,3,YES
      035756  003032                                            .WORD   T$CODE
      035760  036101                                            .WORD   HWPTQ4
      035762  000377                                            .WORD   377
      035764  000001                                            .WORD   T$LOLIM
      035766  000003                                            .WORD   T$HILIM
 8383                   ;INTERRUPT BR LEVEL QUESTION:
 8384 035770                    GPRMD   HWPTQ5,6,0,177400,0,6,YES
      035770  003032                                            .WORD   T$CODE
      035772  036157                                            .WORD   HWPTQ5
      035774  177400                                            .WORD   177400
      035776  000000                                            .WORD   T$LOLIM
      036000  000006                                            .WORD   T$HILIM
```

```
HARDWARE PARAMETER CODING SECTION

 8385
 8386
 8387 036002                                    ENDHRD
                                                                        .EVEN
      036002                                                          L10055:
 8388
 8395
 8396 036002    103     123     122             HWPTQ1: .ASCIZ  /CSR ADDRESS: /
      036005    040     101     104
      036010    104     122     105
      036013    123     123     072
      036016    040     000
 8397 036020    111     116     124             HWPTQ2: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
      036023    105     122     122
      036026    125     120     124
      036031    040     126     105
      036034    103     124     117
      036037    122     040     101
      036042    104     104     122
      036045    105     123     123
      036050    072     040     000
 8398 036053    101     103     124             HWPTQ3: .ASCIZ  /ACTIVE LINE BIT MAP: /
      036056    111     126     105
      036061    040     114     111
      036064    116     105     040
      036067    102     111     124
      036072    040     115     101
      036075    120     072     040
      036100    000
 8399 036101    124     131     120             HWPTQ4: .ASCIZ  /TYPE OF LOOPBACK (1=INTERNAL,2=H3277,3=H325):/
      036104    105     040     117
      036107    106     040     114
      036112    117     117     120
      036115    102     101     103
      036120    113     040     050
      036123    061     075     111
      036126    116     124     105
      036131    122     116     101
      036134    114     054     062
      036137    075     110     063
      036142    062     067     067
      036145    054     063     075
      036150    110     063     062
      036153    065     051     072
      036156    000
 8400 036157    111     116     124             HWPTQ5: .ASCIZ  /INTERRUPT BR LEVEL: /
      036162    105     122     122
      036165    125     120     124
      036170    040     102     122
      036173    040     114     105
      036176    126     105     114
      036201    072     040     000
 8401
 8402                                           .EVEN
```

```
                                        ;*******************************************************************************
*
                                        ;
                                        ;               VDHB.SWQ
                                        ;
                                        ;*******************************************************************************
*


                                        .SBTTL  SOFTWARE PARAMETER CODING SECTION

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

 036204                                         BGNSFT
 036204  000010                                                                         .WORD L10056-L$SOFT/2
 036206                                                                        L$SOFT::

                                        ;UNIT NUMBER PRINTOUT QUESTION:
 036206                                         GPRML   SWPTQ1,0,20,YES
 036206  000130                                                                         .WORD   T$CODE
 036210  036226                                                                         .WORD   SWPTQ1
 036212  000020                                                                         .WORD   20
                                        ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
 036214                                         GPRMD   SWPTQ2,2,D,177777,0,177777,YES
 036214  001052                                                                         .WORD   T$CODE
 036216  036302                                                                         .WORD   SWPTQ2
 036220  177777                                                                         .WORD   177777
 036222  000000                                                                         .WORD   T$LOLIM
 036224  177777                                                                         .WORD   T$HILIM

                                                .EVEN

 036226                                         ENDSFT
                                                                                        .EVEN
 036226                                                                        L10056:


 036226     122     105     120                 SWPTQ1: .ASCIZ  /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /
 036231     117     122     124
 036234     040     125     116
 036237     111     124     040
 036242     116     125     115
 036245     102     105     122
 036250     040     101     123
 036253     040     105     101
 036256     103     110     040
 036261     125     116     111
 036264     124     040     111
 036267     123     040     124
 036272     105     123     124
 036275     105     104     072
 036300     040     000
```

SOFTWARE PARAMETER CODING SECTION

```
 8451 036302  116  125  115        SWPTQ2: .ASCIZ  /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /
      036305  102  105  122
      036310  040  117  106
      036313  040  111  116
      036316  104  111  126
      036321  111  104  125
      036324  101  114  040
      036327  104  101  124
      036332  101  040  105
      036335  122  122  117
      036340  122  123  040
      036343  124  117  040
      036346  122  105  120
      036351  117  122  124
      036354  040  117  116
      036357  040  101  040
      036362  114  111  116
      036365  105  072  040
      036370  000
 8452                             .EVEN
```

SOFTWARE PARAMETER CODING SECTION

```
                                ;*******************************************************************************
                                ;
                                ;               SKL6.P11
                                ;
                                ;*******************************************************************************


036372                          $PATCH::
036372                                  .BLKW   24


                                        LASTAD
                                                                                .EVEN
036442  000000                                                                  .WORD   0
036444  000000                                                                  .WORD   0
036446                          L$LAST::
036446                                  ENDMOD


                                                        .END
```

Symbol table

```
ACTLNS  002234 G        CSRO   = 000000 G        C$RFLA= 000021          EM5401  006705 G        ER8401  012066 G
ADR    = 000020 G       CTRLCF  002264 G         C$RPT  = 000025          EM5402  006744 G        ER9001  012334 G
ADRPTR  013470 G        C$AU   = 000052          C$SEFG= 000046          EM5501  007005 G        ER9002  012424 G
ALTFLD  013000 G        C$AUTO= 000061           C$SPRI= 000041          EM5601  007042 G        ER9101  012572 G
ASLNTL  013052 G        C$BRK  = 000022          C$SVEC= 000037          EM5701  007110 G        ER9301  012620 G
ASSEMB= 000010          C$BSEG= 000004           C$TOME= 000076          EM6401  007156 G        EVL    = 000004 G
BCOUNT  002334 G        C$BSUB= 000002           DELAY   013720 G        EM6402  007205 G        E$END  = 002100
BITTBL  002370 G        C$CLCK= 000062           DFPTBL  002212 G        EM6601  007246 G        E$LOAD= 000035
BIT0   = 000001 G       C$CLEA= 000012           DIAGMC= 000000          EM6602  007274 G        FINACT  014134 G
BIT00  = 000001 G       C$CLOS= 000035           DODMA   013760 G        EM6701  007346 G        F$AU   = 000015
BIT01  = 000002 G       C$CLP1= 000006           DRADRT  002242 G        EM6702  007371 G        F$AUTO= 000020
BIT02  = 000004 G       C$CPBF= 000074           DROP    020040          EM7801  007446 G        F$BGN  = 000040
BIT03  = 000010 G       C$CPME= 000075           EF.CON= 000036 G        EM7802  007501 G        F$CLEA= 000007
BIT04  = 000020 G       C$CVEC= 000036           EF.NEW= 000035 G        EM7901  007532 G        F$DU   = 000016
BIT05  = 000040 G       C$DCLN= 000044           EF.PWR= 000034 G        EM7902  007565 G        F$END  = 000041
BIT06  = 000100 G       C$DODU= 000051           EF.RES= 000037 G        EM8001  007616 G        F$HARD= 000004
BIT07  = 000200 G       C$DRPT= 000024           EF.STA= 000040 G        EM8002  007654 G        F$HW   = 000013
BIT08  = 000400 G       C$DU   = 000053          EF0503  004247 G        EM8101  007720 G        F$INIT= 000006
BIT09  = 001000 G       C$EDIT= 000000           EF0505  004254 G        EM8102  007755 G        F$JMP  = 000050
BIT1   = 000002 G       C$ERDF= 000055           EF1601  004327 G        EM8201  010020 G        F$MOD  = 000000
BIT10  = 002000 G       C$ERHR= 000056           EF3001  004353 G        EM8202  010056 G        F$MSG  = 000011
BIT11  = 004000 G       C$ERRO= 000060           EF3002  004422 G        EM8301  010122 G        F$PROT= 000021
BIT12  = 010000 G       C$ERSF= 000054           EF6401  004471 G        EM8302  010160 G        F$PWR  = 000017
BIT13  = 020000 G       C$ERSO= 000057           EF7801  004546 G        EM8401  010224 G        F$RPT  = 000012
BIT14  = 040000 G       C$ESCA= 000010           EF8401  004604 G        EM8402  010300 G        F$SEG  = 000003
BIT15  = 100000 G       C$ESEG= 000005           EF8402  004676 G        EM8403  010304 G        F$SOFT= 000005
BIT2   = 000004 G       C$ESUB= 000003           EF9001  005013 G        EM8404  010310 G        F$SRV  = 000010
BIT3   = 000010 G       C$ETST= 000001           EF9002  005075 G        EM8405  010313 G        F$SUB  = 000002
BIT4   = 000020 G       C$EXIT= 000032           EF9003  005147 G        EM8406  010317 G        F$SW   = 000014
BIT5   = 000040 G       C$FREQ= 000101           EF9004  005176 G        EM8501  010323 G        F$TEST= 000001
BIT6   = 000100 G       C$FRME= 000100           EF9005  005226 G        EM8502  010377 G        GETPRM  017502
BIT7   = 000200 G       C$GETB= 000026           EF9006  005257 G        EM8601  010403 G        GPRSOB  002430 G
BIT8   = 000400 G       C$GETW= 000027           EF9019  005276 G        EM8602  010464 G        G$CNTO= 000200
BIT9   = 001000 G       C$GMAN= 000043           EF9301  005315 G        EM8701  010554 G        G$DELM= 000372
BMPCQB  002444 G        C$GPHR= 000042           EF9302  005373 G        EM8702  010614 G        G$DISP= 000003
BMPCQE  002644 G        C$GPRI= 000040           EM0101  014570 G        EM8703  010643 G        G$EXCP= 000400
BMPCQP  002442 G        C$INIT= 000011           EM0102  014654 G        EM9009  010646 G        G$HILI= 000002
BOE    = 000400 G       C$INLP= 000020           EM0103  005473 G        EM9010  010672 G        G$LOLI= 000001
BRLEVL  002237 G        C$MANI= 000050           EM0525  005531 G        EM9017  010716 G        G$NO   = 000000
BUFBAS  002704 G        C$MAP  = 000102          EM0526  005621 G        EM9026  011027 G        G$OFFS= 000400
BUFEND  003704 G        C$MEM  = 000031          EM1601  005711 G        EM9104  011053 G        G$OFSI= 000376
BUFMID  003304 G        C$MMU  = 000103          EM4001  005774 G        EM9301  011127 G        G$PRMA= 000001
BUFPTR  002262 G        C$MSG  = 000023          EM4002  006017 G        EM9302  011147 G        G$PRMD= 000002
BUF3QT  003504 G        C$OPNR= 000034           EM4101  006053 G        EM9303  011177 G        G$PRML= 000000
CALMSL  013162 G        C$OPNW= 000104           EM4102  006076 G        EM9304  011244 G        G$RADA= 000140
CHKBMP  013406 G        C$PNTB= 000014           EM4103  006132 G        ENDETB  003704 G        G$RADB= 000000
CKTRAP  013456 G        C$PNTF= 000017           EM4901  006216 G        ENDIT   017722          G$RADD= 000040
CLKBRL  002320 G        C$PNTS= 000016           EM4902  006250 G        ERLTBL  002704 G        G$RADL= 000120
CLKCSR  002316 G        C$PNTX= 000015           EM5001  006302 G        ERRBLK  004112 G        G$RADO= 000020
CLKHRZ  002324 G        C$PUTB= 000072           EM5101  006332 G        ERRMSG  004110 G        G$XFER= 000004
CLKINT  016772 G        C$PUTW= 000073           EM5102  006360 G        ERRNBR  004106 G        G$YES  = 000010
CLKVEC  002322 G        C$QIO  = 000377          EM5103  006416 G        ERRTYP  004104 G        HELP   = 000000
CLNRST  013506 G        C$RDBU= 000007           EM5201  006446 G        ER0101  011320 G        HOE    = 100000 G
CLR16W  013530 G        C$REFG= 000047           EM5202  006472 G        ER0503  011636 G        HWPTQ1  036002
CMPMST  013552 G        C$REL  = 000077          EM5301  006530 G        ER1603  011662 G        HWPTQ2  036020
CONMAP  013644 G        C$RESE= 000033           EM5302  006555 G        ER6401  011742 G        HWPTQ3  036053
CSRA    002242 G        C$REVI= 000004           EM5303  006634 G        ER7801  012040 G        HWPTQ4  036101
```

Symbol table

```
HWPTQ5  036157          L$DVTY  004154 G        L10026  022162          PAR1A   002352 G        SWPTQ1  036226
IBE    = 010000 G       L$EF    002052 G        L10027  022722          PAR2A   002354 G        SWPTQ2  036302
IDU    = 000040 G       L$ENVI  002044 G        L10030  023316          PAR3A   002356 G        S$LSYM= 010000
IER    = 020000 G       L$ERRT  004104 G        L10031  023740          PAR4A   002360 G        TIMER1  002326 G
IESTAT  002270 G        L$ETP   002102 G        L10032  024230          PAR5A   002362 G        TIMER2  002330 G
INDATP  014214 G        L$EXP1  002046 G        L10033  024532          PAR6A   002364 G        TIMER3  002332 G
INDTPX  014244 G        L$EXP4  002064 G        L10034  025224          PAR7A   002366 G        TNUM   = 000032 G
ISR    = 000100 G       L$EXP5  002066 G        L10035  025714          PASCNT  002276 G        TP4FLG  002312 G
IXE    = 004000 G       L$HARD  035724 G        L10036  026360          PCSLOT= 000016 G        TP4RTN  017042 G
I$AU   = 000041          L$HIME  002120 G        L10037  027004          PNT    = 001000 G       TP4VEC  002310 G
I$AUTO= 000041           L$HPCP  002016 G        L10040  027504          PREGRT  004136 G        TSABRT  016002 G
I$CLN  = 000041          L$HPTP  002022 G        L10041  030432          PREG05  004114          TSTNUM  002266 G
I$DU   = 000041          L$HW    002212 G        L10042  031132          PRI    = 002000 G       TXAD1A  002254 G
I$HRD  = 000041          L$ICP   002104 G        L10043  031632          PRI00  = 000000 G       TXAD10= 000012 G
I$INIT= 000041           L$INIT  017100 G        L10044  032246          PRI01  = 000040 G       TXAD2A  002256 G
I$MOD  = 000041          L$LADP  002026 G        L10045  032662          PRI02  = 000100 G       TXAD20= 000014 G
I$MSG  = 000041          L$LAST  036446 G        L10046  033276          PRI03  = 000140 G       TXBFCA  002260 G
I$PROT= 000040           L$LOAD  002100 G        L10047  033712          PRI04  = 000200 G       TXBFCO= 000016 G
I$PTAB= 000041           L$LUN   002074 G        L10050  034306          PRI05  = 000240 G       TXCHA   002244 G
I$PWR  = 000041          L$MREV  002050 G        L10051  034702          PRI06  = 000300 G       TXCHRO= 000002 G
I$RPT  = 000041          L$NAME  002000 G        L10052  035220          PRI07  = 000340 G       TXDATP  016114 G
I$SEG  = 000041          L$PRIO  002042 G        L10053  035640          PRTLPR  014752 G        TXDSBL  016136 G
I$SETU= 000041           L$PROT  017072 G        L10054  035720          PUFIFO  015034 G        TXENBL  016232 G
I$SFT  = 000041          L$PRT   002112 G        L10055  036002          PUFIFR  015116 G        TXIEO   016326 G
I$SRV  = 000041          L$REPP  002062 G        L10056  036226          RBUFA   002244 G -      TXINTC  002304 G
I$SUB  = 000041          L$REV   002010 G        MAPCNT  014342 G        RBUFO  = 000002 G       TXINTF  002306 G
I$TST  = 000041          L$RPT   017064 G        MAPLNS= 000377 G        READBX  015310 G        TXRLNB  004044 G
J$JMP  = 000167          L$SOFT  036206 G        MFUNIT  004216 G        RESETT  015372 G        TXRLNE  004064 G
LGRP1M  002272 G         L$SPC   002056 G        MMENAB  002346 G        RXBDTX= 000030 G        TXRXLB  004004 G
LGRP2M  002274 G         L$SPCP  002020 G        MMPRES  002344 G        RXBETX= 000020 G        TXRXLE  004044 G
LINBIT  014314 G         L$SPTP  002024 G        MMSRO   002342 G        RXBFUL= 000100 G        TXVECA  002232 G
LNCTRA  002252 G         L$STA   002030 G        MSG1    011424 G        RXCNTB  003744 G        T$ARGC= 000002
LNCTRO= 000010 G         L$SW    002224 G        MSG2    011502 G        RXIEO   015504 G        T$CODE= 001052
LOE    = 040000 G        L$TEST  002114 G        MSG3    011561 G        RXINTC  002300 G        T$ERRN= 022125
LOPBCK  002236 G         L$TIML  002014 G        MSLCNT  002340 G        RXINTF  002302 G        T$EXCP= 000000
LOT    = 000010 G        L$UNIT  002012 G        MSLGET  014374 G        RXVECA  002230 G        T$FLAG= 000050
LPCSLT= 000036 G         L10000  002222          MSLOOP  014510 G        ROSLOT= 000002 G        T$GMAN= 000000
LPRA    002246 G         L10001  002230          MSTICK  002336 G        R1SLOT= 000004 G        T$HILI= 177777
LPRO   = 000004 G        L10002  011422          NDERPT  002226 G        R2SLOT= 000006 G        T$LAST= 000001
L$ACP   002110 G         L10003  011660          NEWPAS  017462          R3SLOT= 000010 G        T$LOLI= 000000
L$APT   002036 G         L10004  011740          NEWRES  017454          R4SLOT= 000012 G        T$LSYM= 010000
L$AU    020120 G         L10005  012036          NEWSTA  017144          R5SLOT= 000014 G        T$LTNO= 000032
L$AUT   002070 G         L10006  012064          NUMLNS= 000010 G        SAVBMP  015544 G        T$NEST= 177777
L$AUTO  017752 G         L10007  012332          OOPS    014524 G        SAVMST  015612 G        T$NSO  = 000000
L$CCP   002106 G         L10010  012422          OPTION  002224 G        SETPAR  015656 G        T$NS1  = 000005
L$CLEA  017754 G         L10011  012570          O$APTS= 000000          SFPTBL  002224 G        T$PTNU= 000000
L$CO    002032 G         L10012  012616          O$AU   = 000000         SKPSTS  015724 G        T$SAVL= 177777
L$DEPO  002011 G         L10013  012776          O$BGNR= 000001          STATA   002250 G        T$SEGL= 177777
L$DESC  004164 G         L10014  017070          O$BGNS= 000001          STATO  = 000006 G       T$SUBN= 000000
L$DESP  002076 G         L10016  017750          O$DU   = 000001         STGTRB  004064 G        T$TAGL= 177777
L$DEVP  002060 G         L10017  017752          O$ERRT= 000001          STSTB   002644 G        T$TAGN= 010057
L$DISP  002124 G         L10020  020010          O$GNSW= 000001          STSTE   002704 G        T$TEMP= 000000
L$DLY   002116 G         L10021  020116          O$POIN= 000001          SVCGBL= 000000          T$TEST= 000032
L$DTP   002040 G         L10022  020124          O$SETU= 000000          SVCINS= 000001          T$TSTM= 177777
L$DTYP  002034 G         L10023  020414          PARATB  002350 G        SVCSUB= 000001          T$TSTS= 000001
L$DU    020012 G         L10024  021002          PARATE  002370 G        SVCTAG= 000001          T$$AU  = 010022
L$DUT   002072 G         L10025  021422          PAROA   002350 G        SVCTST= 000001          T$$AUT= 010017
```

J16

Symbol table

```
T$$CLE= 010020          T1      020126 G        T19     032250 G        T4      021424 G        WAIBIS  016576 G
T$$DU = 010021          T10     024534 G        T2      020416 G        T5      022164 G        WAITTX  016652 G
T$$HAR= 010055          T11     025226 G        T20     032664 G        T6      022724 G        WORD1   002314 G
T$$HW = 010000          T12     025716 G        T21     033300 G        T7      023320 G        WTWLNC  016712 G
T$$INI= 010016          T13     026362 G        T22     033714 G        T8      023742 G        WTWLPR  016742 G
T$$MSG= 010013          T14     027006 G        T23     034310 G        T9      024232 G        X$ALWA= 000000
T$$PRO= 010015          T15     027506 G        T24     034704 G        UAM   = 000200 G        X$FALS= 000040
T$$RPT= 010014          T16     030434 G        T25     035222 G        UNITN   002240 G        X$OFFS= 000400
T$$SOF= 010056          T17     031134 G        T26     035642 G        UNSDIV  016366 G        X$TRUE= 000020
T$$SW = 010001          T18     031634 G        T3      021004 G        WAIBIC  016522 G        $PATCH  036372 G
T$$TES= 010054

. ABS.  036446    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 258
Work  file writes: 272
Size of work file: 35960 Words  ( 141 Pages)
Size of core pool: 19714 Words  ( 75 Pages)
Operating  system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:05:41.88
CVDHBD.BIN,CVDHBD.LST/-SP=SVC40/ML,CVDHBD.P11
```