DHV11

DHV-11 FUNC TST PART1

CVDHABO

AH-T653B-MC

FICHE 1 OF 2

JAN 1984

digital

Made In USA

DHV11 DHV-11 FUNC TST PART1
CVDHABO

AH-T653B-MC
FICHE 2 OF 2

JAN 1984
digital
Made In USA

```
.REM &
```

IDENTIFICATION

PRODUCT CODE:	AC-T652B-MC
PRODUCT NAME:	CVDHABO DHV-11 FUNC TST PART1
PRODUCT DATE:	09 OCTOBER 1983
MAINTAINER:	EDSME - DIAGNOSTICS GROUP
AUTHOR:	BERT KLEINSCHMIDT TONY GRIMSHAW
MODIFIED BY:	BERT KLEINSCHMIDT ANTHONY HART

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

ORIGINAL RELEASE:       31-OCT-83 (EDITED 11-JUL-83)
                        BERT KLEINSCHMIDT

VERSION B0              09-OCT-83       BERT KLEINSCHMIDT
        FIXED TYPOGRAPHICAL ERRORS.
        MOVED TEST FROM CVDHB (PART 2) INTO THIS PROGRAM:
                OLD CVDHB (VERSION A) TESTS 2 THROUGH 8 ARE
                NOW NEW CVDHA (VERSION B) TESTS 20 THROUGH 26.
        MODIFIED TEST 13 (SELFTEST FAIL TEST) TO ALLOW USE OF CVDHA
                IN MANUFACTURING WITH VERSION 0 DHV ROM CODE.
```

TABLE OF CONTENTS

1.0 GENERAL PROGRAM CONSIDERATIONS
1.1 PROGRAM ABSTRACT
1.2 SYSTEM REQUIREMENTS
1.3 RELATED DOCUMENTS AND STANDARDS
1.4 DIAGNOSTIC HIERARCY PREREQUISITES
2.0 OPERATING INSTRUCTIONS
2.1 COMMANDS
2.2 SWITCHES
2.3 FLAGS
2.4 EXTENDED COMMAND SYNTAX
2.4.1 START COMMAND
2.4.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)
2.4.1.2 PASS SWITCH (/PASS:<PASS-CNT>)
2.4.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)
2.4.1.4 END OF PASS SWITCH (/EOP:<INCR>)
2.4.1.5 EFFECT OF START COMMAND
2.4.2 RESTART COMMAND
2.4.2.1 TESTS, PASS, AND FLAGS SWITCHES
2.4.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)
2.4.2.3 EFFECT OF RESTART COMMAND
2.4.3 CONTINUE COMMAND
2.4.3.1 FLAG SWITCH (/FLAGS:<FLAG-LIST>)
2.4.3.2 EFFECT OF CONTINUE COMMAND
2.4.4 PROCEED COMMAND
2.4.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)
2.4.4.2 EFFECT OF PROCEED COMMAND
2.4.5 ADD COMMAND
2.4.6 EFFECT OF ADD COMMAND
2.4.7 DROP COMMAND
2.4.8 EFFECT OF DROP COMMAND
2.4.9 PRINT COMMAND
2.4.9.1 EFFECT OF PRINT COMMAND
2.4.10 DISPLAY COMMAND
2.4.10.1 EFFECT OF DISPLAY COMMAND
2.4.11 FLAGS COMMAND
2.4.11.1 EFFECT OF FLAGS COMMAND
2.4.12 ZFLAGS COMMAND
2.4.13 ZFLAGS COMMAND
2.4.14 CONTROL CHARACTERS
2.5 HARDWARE QUESTIONS
2.6 SOFTWARE QUESTIONS
2.7 EXTENDED P-TABLE DIALOGUE
2.8 QUICK START-UP PROCEDURE (XXDP+)
3.0 ERROR INFORMATION
3.1 TYPES OF ERROR MESSAGES
3.2 ERROR MESSAGES
4.0 PERFORMANCE AND PROGRESS REPORTS
5.0 TEST SUMMARIES
6.0 EXAMPLE ERROR FREE PASS

1.0 GENERAL PROGRAM CONSIDERATIONS

1.1 PROGRAM ABSTRACT

CVDHA IS PART ONE OF THE DHV-11 FUNCTIONAL VERIFICATION TEST. THIS PART OF THE TEST VERIFIES THAT THE RESET, REGISTER ACCESS, AND INTERRUPT FUNCTIONS OF THE BOARD ARE FUNCTIONING CORRECTLY.

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

1.2 SYSTEM REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DHV FVT:

- O LSI-11 PROCESSOR WITH AT LEAST 32 KBYTES OF RAM.
- O DHV11 BOARDS INSTALLED ON THE Q-BUS.
- O APPROPRIATE PROGRAM LOAD DEVICE SUPPORTING XXDP+ MEDIA OR A DOWN-LINE LOADING SYSTEM.

1.3 RELATED DOCUMENTS AND STANDARDS

- O DHV-11 HARDWARE MANUAL - THIS MANUAL DESCRIBES THE FUNCTIONS AND USES OF THE DHV-11 DEVICE.
- O XXDP+ USER'S MANUAL - DESCRIBES THE RUNNING OF DIAGNOSTICS UNDER THE XXDP+ MONITOR.

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE LSI-11 PROCESSOR, THE Q-BUS, THE SYSTEM MEMORY, THE CONSOLE TERMINAL, AND THE LOAD MEDIA ARE ASSUMED TO HAVE BEEN TESTED AND FOUND WORKING BEFORE THIS PROGRAM 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 DHV-11 CAN BE PLACED (160460 OCTAL).

2. INTERRUPT VECTOR ADDRESS - THIS QUESTION REQUESTS THE INTERRUPT VECTOR ADDRESS OF THE SPECIFIED DHV11.

3. ACTIVE LINES BIT MAP - THIS QUESTION REQUESTS AN OCTAL BIT MAP OF THE SERIAL COMMUNICATION LINES ON THE DHV11 WHICH ARE BEING SELECTED FOR TESTING. IF THE BIT IN THE BIT MAP IS SET WHICH CORRESPONDS TO A PARTICULAR LINE (I.E. BIT 3 FOR LINE 3) THAT LINE WILL BE TESTED BY THE FVT.

4. BR LEVEL - THIS QUESTIONS REQUESTS THE INTERRUPT BR LEVEL OF THE DHV11.

2.6 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". THE FOLLOWING SOFTWARE P-TABLE QUESTIONS ARE ASKED BY THE PROGRAM IF THE OPERATOR INDICATES THAT THE SOFTWARE PARAMETERS ARE TO BE CHANGED.

1. REPORT UNIT NUMBER AS EACH UNIT IS TESTED - THIS QUESTION ASKS WHETHER THE PROGRAM SHOULD REPORT THE NUMBER OF THE UNIT WHICH IT IS TESTING AS IT BEGINS TO TEST EACH UNIT.

2. NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE - THIS QUESTION ASKS FOR THE NUMBER OF DATA ERRORS WHICH SHOULD BE REPORTED INDIVIDUALLY BY THIS PROGRAM FOR EACH LINE FOR EACH TRANSMISSION TEST. ERRORS WHICH ARE NOT REPORTED INDIVIDUALLY ARE REPORTED IN SUMMARY ERROR REPORTS.

3. ROM VERSION PRINTOUT ON THE FIRST PASS - THIS QUESTION ASKS WHETHER THE PROGRAM SHOULD PRINT OUT THE VERSIONS OF THE ON BOARD 8051 PROCESSOR ROMS DURING THE FIRST PASS OF THE PROGRAM.

2.7 EXTENDED P-TABLE DIALOGUE

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

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

```
# UNITS (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 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 (O) 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) O ?  1,0<CR>

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

UNIT 7
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  6,7<CR>
Q-FACTOR (O) O ?  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) O ?  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 DHV-11 BOARDS. TO EXECUTE THE PROGRAM IN THIS MODE THE OPERATOR CAN RUN WITH THE INHIBIT BASIC ERROR REPORTING SWITCH. IN THIS MODE THE PROGRAM PRINTS ERROR MESSAGES WHICH CONTAIN THE ERROR MESSAGE HEADER DESCRIBED ABOVE, PLUS THE NAME OF THE FAILING TEST. FOR A LIST OF THE TEST NAMES IN THIS PROGRAM SEE THE TEST SUMMARIES SECTION OF THIS DOCUMENT. AN EXAMPLE OF SUCH AN ERROR MESSAGE IS THE FOLLOWING:

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

THIS ERROR INDICATES THAT A FATAL ERROR WAS ENCOUNTERED WITHIN THE TEST WHICH TESTS THE READ/WRITE CAPABILITY OF THE DHV-11 REGISTERS.

IF THE OPERATOR REQUIRES MORE EXTENSIVE ERROR REPORTING HE CAN RUN WITH ALL ERROR REPORTING ENABLED BY NOT USING THE INHIBIT REPORTING SWITCHES. THE ABOVE ERROR MESSAGE WOULD THEN BECOME THE FOLLOWING:

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

4.0 PERFORMANCE AND PROGRESS REPORTS

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

5.0 TEST SUMMARIES

THE FOLLOWING TESTS ARE INCLUDED WITHIN CVDHA:

1. DEVICE REGISTER ADDRESS TEST - VERIFIES THAT THE UUT REGISTERS WILL RESPOND WITHT THE PROPER Q-BUS HANDSHAKING WHEN ACCESSED. VERIFIES THAT THE UUT IS AT THE PROPER ADDRESS.

2. MASTER.RESET (SELFTEST) TEST - VERIFIES THAT THE MASTER.RESET BIT CLEARS WITHIN A SPECIFIED TIME OF IT BEING SET.

3. MASTER.RESET (SKIP SELFTEST) TEST - VERIFIES THAT THE MASTER.RESET BIT CLEARS WITHIN A SHORT TIME AFTER IT IS SET IF THE SKIP SELFTEST SEQUENCE IS USED.

4. RX.CHARACTER FIELD TEST - VERIFIES THAT THE DATA BITS OF THE CODES IN THE FIFO AFTER A RESET AND SKIP SELFTEST ARE CONSISTANT WITH THE SKIP SELFTEST CODES.

5. RECEPTION FLAG FIELD TEST - VERIFIES THAT THE 3 DATA STATUS BITS (OVERRRUN, FRAMING, AND PARITY ERROR BITS) ARE ALL SET ON ALL OF THE SKIP SELFTEST CODES IN THE FIFO AFTER A RESET AND SKIP SELFTEST SEQUENCE.

6. RX.DATA.AVAIL TEST - VERIFIES THAT THE RX.DATA.AVAIL BIT IS SET WHEN THE SKIP SELFTEST CODES ARE IN THE FIFO AND THAT IT CLEARS AFTER THEY ARE READ.

7. RX.DATA.VALID TEST - VERIFIES THAT THE RX.DATA.VALID BIT IS SET FOR EACH VALID SKIP SELFTEST CODE IN THE FIFO AND CLEAR AFTER ALL VALID CODES ARE READ.

8. RX.LINE FIELD TEST - VERIFIES THAT THE RX.LINE FIELDS ARE CORRECT FOR THE SKIP SELFTEST CODES.

9. BMP RUN TEST - THIS TEST RUNS THE BMP AND VERIFIES THAT IT DOES NOT FAIL WITHIN A SPECIFIED PERIOD. THIS TEST SHOULD SIGNAL PROBLEMS THAT THE BMP CODES COULD CAUSE WITH LATER TESTS.

10. SKIP SELFTEST TEST - THIS TEST VERIFIES THAT IF THE SELFTEST IS SKIPPED THE PROPER CODES ARE PLACED IN THE FIFO AND THAT NO ERRORS ARE ENCOUNTERED.

11. DIAGNOSTIC.FAIL (SKIP SELFTEST) TEST - THIS TEST VERIFIES, BY USING THE SKIP SELFTEST SEQUENCE, THAT THE DIAGNOSTIC.FAIL BIT WILL GO TO BOTH THE ACTIVE AND INACTIVE STATES.

12. SELFTEST RUN TEST - VERIFIES THAT NO ERRORS ARE FOUND BY THE EXECUTION OF THE SELFTEST.

13. SELFTEST FAIL TEST - VERIFIES THAT THE SELFTEST WILL REPORT ERRORS CORRECTLY WHEN IT IS FORCED TO FAIL.

14. ROM VERSION PRINTOUT TEST - IF REQUESTED, REPORTS THE VERSION NUMBERS OF THE 8051 ROMS.

15. WORD ACCESS READ/WRITE TEST - VERIFIES THAT THE REGISTERS RESPOND CORRECTLY TO WORD READ AND WRITE ACCESSES.

16. WORD ACCESS READ/MODIFY/WRITE TEST - VERIFIES THAT THE REGISTERS RESPOND CORRECTLY TO READ/MODIFY/WRITE WORD ACCESSES.

17. BYTE ACCESS READ/WRITE TEST - VERIFIES THAT THE REGISTERS RESPOND CORRECTLY TO BYTE READ AND WRITE ACCESSES.

18. BYTE ACCESS READ/MODIFY/WRITE TEST - VERIFIES THAT THE REGISTERS RESPOND CORRECTLY TO READ/MODIFY/WRITE BYTE ACCESSES.

19. ID.BIT TEST - VERIFIES THAT THE ID.BIT READS AS A ZERO.

20. NO TX.DATA.VALID/NO TX.ACTION TEST - VERIFIES THAT IF A DATA WORD IS WRITTEN WITHOUT THE TX.DATA.VALID BIT SET, NO TX.ACTION IS GENERATED. THIS TEST DOES NOT REQUIRE THAT CHARACTERS ARE TXED.

21. TX.DATA.VALID / TX.ACTION TEST - VERIFIES THAT IF A DATA WORD IS WRITTEN WITH THE TX.DATA.VALID BIT SET, IT GENERATES A CORRESPONDING TX.ACTION. THIS TEST DOES NOT REQUIRE THAT CHARACTERS ARE TXED.

22. TX.ENABLE INACTIVE TEST - VERIFIES THAT IF THE TX.ENABLE BIT IS CLEAR NO TRANSMISSION OCCURS.

23. TX.ENABLE ACTIVE TEST - VERIFIES THAT TX OCCURS IF THE TX.ENABLE IS SET.

24. INTERRUPTS TEST - VERIFIES THAT THE TX AND RX INTERRUPTS ARE FUNCTIONING CORRECTLY.

25. BR LEVEL TEST - VERIFIES THAT THE UUT GENERATES TX AND RX INTERRUPTS AT THE CORRECT BR LEVEL.

26. DIAG FIELD (BMP) TEST - VERIFIES THAT A REQUEST FOR BMP CODE REPORTING IS ANSWERED BY THE UUT WITHIN THE SPECIFIED TIME.

27. REPORT BMP CODES TEST - THIS PSEUDO TEST REPORTS THE FIRST 32 BMP CODES WHICH WERE DISCOVERED IN THE FIFO DURING THE EXECUTION OF THE OTHER TESTS. THIS AVOIDS THE INTERRUPTION OF OTHER TESTS BY THESE CODES, IF THEY ARE NOT CRITICAL TO THE TESTS BEING PERFORMED.

6.0 EXAMPLE ERROR FREE PASS

THE FOLLOWING IS AN EXAMPLE OF AN ERROR FREE PASS DIALOGUE:

```
.R CVDHABO
CVDHABO.BIC

DRS
CVDHA-B-0
DHV-11 FUNC TST PART1
UNIT IS DHV-11
RESTART ADDR: 147670
DR>STA

CHANGE HW (L)  ? Y

# UNITS (D)  ? 2

UNIT 0
CSR ADDRESS:  (O)  160460 ? ↑Z

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

CHANGE SW (L)  ? Y

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

 TESTING UNIT :  0(D)

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

 TESTING UNIT :  1(D)

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

CVDHA EOP    1
    0 CUMULATIVE ERRORS

 TESTING UNIT :  0(D)
↑C
```

```
DR> EXIT

.

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


                                ;*****************************************************************************
                                ;
                                ;               FVTA.PHD
                                ;
                                ;*****************************************************************************


                                .SBTTL  PROGRAM HEADER


                                        .MCALL  SVC
000000                                  SVC                                     ; INITIALIZE SUPERVISOR MACROS

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

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

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

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

002000                          BGNMOD

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

002000                          POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL


                                        HEADER  CVDHA,B,0,16,0,PRIO7
002000                                                          L$NAME::
002000  103                                                             .ASCII /C/
002001  126                                                             .ASCII /V/
002002  104                                                             .ASCII /D/
```

```
002003      110                                   .ASCII /H/
002004      101                                   .ASCII /A/
002005      000                                   .BYTE  0
002006      000                                   .BYTE  0
002007      000                                   .BYTE  0
002010                                      L$REV::
002010      102                                   .ASCII /B/
002011                                      L$DEPO::
002011      060                                   .ASCII /0/
002012                                      L$UNIT::
002012  000000                                    .WORD  0
002014                                      L$TIML::
002014  000016                                    .WORD  16
002016                                      L$HPCP::
002016  036662                                    .WORD  L$HARD
002020                                      L$SPCP::
002020  037054                                    .WORD  L$SOFT
002022                                      L$HPTP::
002022  002214                                    .WORD  L$HW
002024                                      L$SPTP::
002024  002226                                    .WORD  L$SW
002026                                      L$LADP::
002026  037372                                    .WORD  L$LAST
002030                                      L$STA::
002030  000000                                    .WORD  0
002032                                      L$CO::
002032  000000                                    .WORD  0
002034                                      L$DTYP::
002034  000000                                    .WORD  0
002036                                      L$APT::
002036  000000                                    .WORD  0
002040                                      L$DTP::
002040  002124                                    .WORD  L$DISPATCH
002042                                      L$PRIO::
002042  000340                                    .WORD  PRIO7
002044                                      L$ENVI::
002044  000000                                    .WORD  0
002046                                      L$EXP1::
002046  000000                                    .WORD  0
002050                                      L$MREV::
002050      003                                   .BYTE  C$REVISION
002051      003                                   .BYTE  C$EDIT
002052                                      L$EF::
002052  000000                                    .WORD  0
002054  000000                                    .WORD  0
002056                                      L$SPC::
002056  000000                                    .WORD  0
002060                                      L$DEVP::
002060  004036                                    .WORD  L$DVTYP
002062                                      L$REPP::
002062  024232                                    .WORD  L$RPT
002064                                      L$EXP4::
002064  000000                                    .WORD  0
002066                                      L$EXP5::
002066  000000                                    .WORD  0
002070                                      L$AUT::
002070  000000                                    .WORD  0
```

PROGRAM HEADER

```
002072
002072  025120
002074
002074  000000
002076
002076  004046
002100
002100  104035
002102
002102  003766
002104
002104  024246
002106
002106  025102
002110
002110  025100
002112
002112  024240
002114
002114  000000
002116
002116  000000
002120
002120  000000
1113

L$DUT::
        .WORD   L$DU
L$LUN::
        .WORD   0
L$DESP::
        .WORD   L$DESC
L$LOAD::
        EMT     E$LOAD
L$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
```

```
                        .SBTTL  DISPATCH TABLE

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

                                DISPATCH 27
002122                                                          .WORD   27
002122  000033
002124                                                  L$DISPATCH::
002124  025236                                                  .WORD   T1
002126  025526                                                  .WORD   T2
002130  025756                                                  .WORD   T3
002132  026222                                                  .WORD   T4
002134  026420                                                  .WORD   T5
002136  026612                                                  .WORD   T6
002140  027030                                                  .WORD   T7
002142  027236                                                  .WORD   T8
002144  027444                                                  .WORD   T9
002146  027632                                                  .WORD   T10
002150  030046                                                  .WORD   T11
002152  030274                                                  .WORD   T12
002154  030556                                                  .WORD   T13
002156  031134                                                  .WORD   T14
002160  031534                                                  .WORD   T15
002162  031752                                                  .WORD   T16
002164  032174                                                  .WORD   T17
002166  032446                                                  .WORD   T18
002170  032724                                                  .WORD   T19
002172  033040                                                  .WORD   T20
002174  033256                                                  .WORD   T21
002176  033520                                                  .WORD   T22
002200  034026                                                  .WORD   T23
002202  034372                                                  .WORD   T24
002204  035424                                                  .WORD   T25
002206  036314                                                  .WORD   T26
002210  036600                                                  .WORD   T27

```

```
                        ;**********************************************************************
                        ;
                        ;               FVTA.DHT
                        ;
                        ;**********************************************************************
                        
                        
                        .SBTTL  DEFAULT HARDWARE P-TABLE
                        
                        ;++
                        ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                        ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                        ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
                        ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
                        ;--
                        
002212                          BGNHW   DFPTBL
002212  000004                                                         .WORD   L10000-L$HW/2
002214                                                                  L$HW::
002214                                                                  DFPTBL::
                        
002214  160460                  .WORD   160460  ;DEFAULT CSR ADDRESS
002216  000300                  .WORD   300     ;DEFAULT VECTOR ADDRESS
002220  177777                  .WORD   177777  ;DEFAULT ACTIVE LINES BIT MAP
002222     004                  .BYTE   4       ;DEFAULT BR LEVEL
                                .EVEN
                        
002224                          ENDHW
002224                                                                  L10000:
```

```
1169
1170                            ;*****************************************************************************************
1171                            ;
1172                            ;               FVTA.SWT
1173                            ;
1174                            ;*****************************************************************************************
1175
1176
1177
1178                            .SBTTL  SOFTWARE P-TABLE
1179
1180                            ;++
1181                            ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
1182                            ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
1183                            ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
1184                            ; AT RUN TIME.
1185                            ;--
1186
1187 002224                            BGNSW   SFPTBL
     002224 000002                                                          .WORD   L10001-L$SW/2
     002226                                                         L$SW::
     002226                                                         SFPTBL::
1188
1189 002226 000021                      OPTION::        .WORD   21      ;BIT MAP OF PROGRAM CONTROL FLAGS
1190 002230 000000                      NDERPT::        .WORD   0       ;DEFAULT NUMBER OF INDIVIDUAL DATA ERRORS TO RPT.
1191
1192 002232                             ENDSW
     002232                                                         L10001:
```

```
                        ;*****************************************************************************
                        ;
                        ;               FVTA.EQU
                        ;
                        ;*****************************************************************************


                        .SBTTL  GLOBAL EQUATES SECTION




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

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

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

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


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

GLOBAL EQUATES SECTION

```
000001                  BIT00== 1
                        ;
001000                  BIT9==  BIT09
000400                  BIT8==  BIT08
000200                  BIT7==  BIT07
000100                  BIT6==  BIT06
000040                  BIT5==  BIT05
000020                  BIT4==  BIT04
000010                  BIT3==  BIT03
000004                  BIT2==  BIT02
000002                  BIT1==  BIT01
000001                  BIT0==  BIT00
                        ;
                        ; EVENT FLAG DEFINITIONS
                        ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                        ;
000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
                        ;
                        ;
                        ; PRIORITY LEVEL DEFINITIONS
                        ;
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
```

1256

```
                        ;*******************************************************************************
                        ;
                        ;               FVTA.GDT
                        ;
                        ;*******************************************************************************


                        .SBTTL  GLOBAL DATA SECTION

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

                        ;*****************************************************************************
                        ;       UNIT VARIBLE AREA
                        ;*****************************************************************************

002232  000300                  RXVECA:: .WORD  300      ;RX VECTOR ADDRESS.
002234  000304                  TXVECA:: .WORD  304      ;TX VECTOR ADDRESS.
002236  000377                  ACTLNS:: .WORD  377      ;ACTIVE LINE BIT MAP.
002240  000000                  UNITN:: .WORD   0        ;UNIT NUMBER.
002242     004                  BRLEVL:: .BYTE  4        ;INTERRUPT BUS REQUEST LEVEL
                                         .EVEN

                        ;*****************************************************************************
                        ;       DEVICE REGISTER ADDRESS TABLE
                        ;*****************************************************************************
002244                  DRADRT::
002244  160020                  CSRA::   .WORD   160020  ;DHV-11 CSR ADDRESS
002246  160022          TXCHA:: RBUFA::  .WORD   160022  ;DHV-11 RECEIVE/TRANSMIT BUFFER ADDRESS
002250  160024                  LPRA::   .WORD   160024  ;DHV-11 LINE PARAMETER REGISTER ADDRESS
002252  160026                  STATA::  .WORD   160026  ;DHV-11 STATUS REGISTER ADDRESS
002254  160030                  LNCTRA:: .WORD   160030  ;DHV-11 LINE CONTROL REGISTER ADDRESS
002256  160032                  TXAD1A:: .WORD   160032  ;DHV-11 TRANSMIT BUFFER 1 REGISTER ADDRESS
002260  160034                  TXAD2A:: .WORD   160034  ;DHV-11 TRANSMIT BUFFER 2 REGISTER ADDRESS
002262  160036                  TXBFCA:: .WORD   160036  ;DHV-11 TRANSMIT BUFFER COUNT REGISTER ADDRESS


                        ;*****************************************************************************
                        ;       BIT MASK TABLE OF UN-USED DHV DEVICE REGISTER BITS.
                        ;*****************************************************************************
002264  137660          UNBTTB::.WORD   137660  ;UNUSED BIT MASK FOR THE CSR
002266  177777                  .WORD   177777  ;UNUSED BIT MASK FOR THE RBUF/TX REG
002270  000007                  .WORD   7       ;UNUSED BIT MASK FOR THE LPR
002272  177777                  .WORD   177777  ;UNUSED BIT MASK FOR THE STAT
002274  166051                  .WORD   166051  ;UNUSED BIT MASK FOR THE LNCTRL
002276  000000                  .WORD   0       ;UNUSED BIT MASK FOR THE TBUFFAD1
002300  077700                  .WORD   77700   ;UNUSED BIT MASK FOR THE TBUFFAD2
002302  000000                  .WORD   0       ;UNUSED BIT MASK FOR THE TBUFFCT

                        ;*****************************************************************************
                        ;       REGISTER MESSAGE ADDRESS TABLE
                        ;*****************************************************************************
002304  006002                  RMATBB:: .WORD  DROOMG  ;ADDRESS OF "CSR" MESSAGE.
```

```
1315 002306  006006                    .WORD   DRO2MG  ;ADDRESS OF "RBUF" MESSAGE.
1316 002310  006013                    .WORD   DRO4MG  ;ADDRESS OF "LPR" MESSAGE.
1317 002312  006017                    .WORD   DRO6MG  ;ADDRESS OF "STAT" MESSAGE.
1318 002314  006024                    .WORD   DR10MG  ;ADDRESS OF "LNCTRL" MESSAGE.
1319 002316  006033                    .WORD   DR12MG  ;ADDRESS OF "TBUFFAD1" MESSAGE.
1320 002320  006044                    .WORD   DR14MG  ;ADDRESS OF "TBUFAD2" MESSAGE.
1321 002322  006055                    .WORD   DR16MG  ;ADDRESS OF "TBUFFCT" MESSAGE.
1322
1323                            ;*******************************************************************************
1324                            ;       ASSORTED GLOBAL VARIABLES:
1325                            ;*******************************************************************************
1326 002324  000000                    BUFPTR:: .WORD  0       ;STORAGE FOR RECEIVE CHARACTER BUFFER POINTER.
1327 002326  000001                    TSTNUM:: .WORD  1       ;STORAGE FOR THE TEST NUMBER.
1328 002330  000000                    IESTAT:: .WORD  0       ;STORAGE FOR THE INTERRUPT ENABLE BIT STATES.
1329 002332  000000                    PASCNT:: .WORD  0       ;STO'G FOR PASS COUNT USED IN ROM VERSION# TST.
1330 002334  000000                    RXINTC:: .WORD  0       ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
1331 002336  000000                    RXINTF:: .WORD  0       ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
1332 002340  000000                    TXINTC:: .WORD  0       ;STORAGE FOR TRANSMIT INTERRUPT COUNT.
1333 002342  000000                    TXINTF:: .WORD  0       ;STORAGE FOR TRANSMIT INTERRUPT FLAGS.
1334 002344  000000                    TP4VEC:: .WORD  0       ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
1335 002346  000000                    TP4FLG:: .WORD  0       ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
1336 002350  000000                    WORD1::  .WORD  0       ;LOCATION FOR PASSING INDIRECT PARAMETERS.
1337 002352  000000                    CTRLCF:: .WORD  0       ;STORAGE FOR THE CONTROL-C FLAG.
1338
1339                            ;*******************************************************************************
1340                            ;       LINE TIME CLOCK VARIABLES AND STORAGE.
1341                            ;*******************************************************************************
1342 002354  177546                    CLKCSR:: .WORD  177546  ;CSR ADDRESS OF THE LTC.
1343 002356  000300                    CLKBRL:: .WORD  PRI06   ;INTERRUPT PRIORITY LEVEL OF THE LTC.
1344 002360  000100                    CLKVEC:: .WORD  100     ;INTERRUPT VECTOR ADDRESS OF THE LTC.
1345 002362  000074                    CLKHRZ:: .WORD  60.     ;INTERRUPT FREQUENCY OF THE LTC.
1346 002364  000000                    TIMER1:: .WORD  0       ;HARDWARE CLOCK COUNTER #1.
1347 002366  000000                    TIMER2:: .WORD  0       ;HARDWARE CLOCK COUNTER #2.
1348 002370  000170                    TIMER3:: .WORD  120.    ;HARDWARE BREAK COUNTER LOCATION.
1349 002372  000170                    BCOUNT:: .WORD  120.    ;BREAK COUNT VALUE IN CLOCK TICKS.
1350 002374  000021                    MSTICK:: .WORD  17.     ;NUMBER OF MILLI-SECONDS PER LTC TICK.
1351 002376  000062                    MSLCNT:: .WORD  62      ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.
1352
1353                            ;*******************************************************************************
1354                            ;       MEMMORY MANAGEMENT VARIABLES AND FLAGS.
1355                            ;*******************************************************************************
1356 002400  177572                    MMSR0:: .WORD   177572  ;ADDRESS OF MEM MGT STATUS REGISTER #0.
1357 002402  000000                    MMPRES:: .WORD  0       ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
1358 002404  000000                    MMENAB:: .WORD  0       ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).
1359 002406  172340                    PAR0A:: .WORD   172340  ;ADDRESS OF MEM MGT PAR #0.
1360
1361                            ;*******************************************************************************
1362                            ;       TABLE OF WORDS WITH CORRESPONDING BIT SET FOR GENERATION OF BIT MAPS.
1363                            ;*******************************************************************************
1364 002410  000001            BITTBL:: .WORD  1               ;BIT 0 SET.
1365 002412  000002                     .WORD  2               ;BIT 1 SET.
1366 002414  000004                     .WORD  4               ;BIT 2 SET.
1367 002416  000010                     .WORD  10              ;BIT 3 SET.
1368 002420  000020                     .WORD  20              ;BIT 4 SET.
1369 002422  000040                     .WORD  40              ;BIT 5 SET.
1370 002424  000100                     .WORD  100             ;BIT 6 SET.
1371 002426  000200                     .WORD  200             ;BIT 7 SET.
```

```
1372 002430  000400                    .WORD   400             ;BIT 8 SET.
1373 002432  001000                    .WORD   1000            ;BIT 9 SET.
1374 002434  002000                    .WORD   2000            ;BIT 10 SET.
1375 002436  004000                    .WORD   4000            ;BIT 11 SET.
1376 002440  010000                    .WORD   10000           ;BIT 12 SET.
1377 002442  020000                    .WORD   20000           ;BIT 13 SET.
1378 002444  040000                    .WORD   40000           ;BIT 14 SET
1379 002446  100000                    .WORD   100000          ;BIT 15 SET.
1380
1381                            ;*****************************************************************************
1382                            ;*      GPR SAVE AREA ZERO.
1383                            ;*****************************************************************************
1384 002450                     GPRS0B::                        ;BASE OF GPR SAVE AREA NUMBER ZERO.
1385 002450  000000                    .WORD   0               ;WORD 1, STORAGE FOR R1.
1386 002452  000000                    .WORD   0               ;WORD 2, STORAGE FOR R2.
1387 002454  000000                    .WORD   0               ;WORD 3, STORAGE FOR R3.
1388 002456  000000                    .WORD   0               ;WORD 4, STORAGE FOR R4.
1389 002460  000000                    .WORD   0               ;WORD 5, STORAGE FOR R5.
1390
1391                            ;*****************************************************************************
1392                            ;*      TRANSMISSION AND RECEPTION VARIABLES, POINTERS, AND FLAGS.
1393                            ;*****************************************************************************
1394 002462  000000                    ERSMRF:: .WORD  0       ;ERROR SUMMARY REPORT FLAGS.
1395 002464                            ERCNTB:: .BLKW  16.     ;TABLE OF ERROR COUNTERS.
1396
1397                            ;*****************************************************************************
1398                            ;       STORAGE AREA FOR THE BMP CODE QUEUE.
1399                            ;*****************************************************************************
1400 002524  000000             BMPCQP::        .WORD   0       ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
1401 002526                     BMPCQB::        .BLKW   64.     ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
1402 002726                     BMPCQE::                        ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1403                            ;*****************************************************************************
1404                            ;       GENERAL TABLE AND BUFFER AREA--513 WORDS.
1405                            ;*****************************************************************************
1406 002726                     BUFBAS::                        ;BASE OF MEMORY BUFFER.
1407 002726                     ERLTBL::        .BLKW   128.    ;FIRST HALF OF GENERAL TABLE OR BUFFER.
1408 003326                     BUFMID::        .BLKW   64.     ;SECOND HALF OF GENERAL TABLE OR BUFFER.
1409 003526                     BUF3QT::        .BLKW   64.     ;LAST QUARTER OF THE BUFFER AREA.
1410 003726                     BUFEND::                        ;END OF GENERAL PURPOSE MEMORY BUFFER.
1411 003726                     ENDETB::        .BLKW   16.     ;BUFFER OVERFLOW SPACE.
1412
1425 003766                             ERRTBL
     003766                                                                 L$ERRTBL::
     003766  000000             ERRTYP::        .WORD   0
     003770  000000             ERRNBR::        .WORD   0
     003772  000000             ERRMSG::        .WORD   0
     003774  000000             ERRBLK::        .WORD   0
1426
1427                                    .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 SUBROUTINE WHICH CALLED PREG05.
                        ;*
                        ;*      EACH LEVEL OF SUBROUTINE 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.
                        ;*******************************************************************************
```

```
                                .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 PREGOS
                                ;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.
```

```
1483                            .SBTTL  GLOBAL MACRO DEFINITION          - SAVE -
1484                            ;*****************************************************************************
1485                            ;*      THIS MACRO IS USED AT THE BEGINNING OF A SUBROUTINE TO SAVE THE
1486                            ;*      CONTENTS OF THE GPRS R0 THRU R5.
1487                            ;*
1488                            ;* INPUTS:      SP - UNCHANGED SINCE SUBROUTINE WAS ENTERED
1489                            ;*              R5SLOT - OFFSET TO STACK SLOT FOR R5 (EQUATED TO 14 OCTAL)
1490                            ;*
1491                            ;* OUTPUTS:     GPR SAVE AREA ON THE STACK IS LOADED WITH THE CONTENTS OF GPRS
1492                            ;*              TOP OF STACK - LOADED WITH THE RETURN ADDRESS INTO PREG05
1493                            ;*
1494                            ;* CALLING SEQUENCE:    SAVE
1495                            ;*
1496                            ;* COMMENTS:    NO ARGUMENTS ARE ALLOWED.
1497                            ;*              THE PASS MACRO SHOULD BE CALLED TO RESTORE THE GPR VALUES.
1498                            ;*
1499                            ;* SUBORDINATE ROUTINES CALLED: PREG05.
1500                            ;*****************************************************************************
1501
1502                                  .MACRO  SAVE
1503                                  .LIST
1504                                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
1505                                  .NLIST
1506                                  .ENDM   SAVE
```

```
1508                    .SBTTL  GLOBAL MACRO DEFINITION          - PASS -
1509                    ;*****************************************************************************
1510                    ;*      THIS MACRO IS USED IN CONJUNCTION WITH THE SAVE MACRO.  IT IS
1511                    ;*      CALLED AT END OF A SUBROUTINE TO PASS PARAMETERS IN GPRS BACK TO THE
1512                    ;*      CALLING ROUTINE BY ALTERING THE GPR SAVE AREA ON THE STACK AND THEN
1513                    ;*      RETURNING TO PREGO5 TO RESTORE THE GPRS TO THEIR SAVED VALUES.
1514                    ;*
1515                    ;* INPUTS:      ONLY ALLOWED ARGUMENTS ARE "R0" THRU "R5".
1516                    ;*              R0SLOT THRU R5SLOT MUST BE EQUATED TO THEIR RESPECTIVE GPR SAVE
1517                    ;*                      SLOT OFFSETS BEFORE CALLING THIS MACRO.
1518                    ;*
1519                    ;* OUTPUTS:     THE GPR VALUES ARE PUT IN THEIR RESPECTIVE SLOTS ON THE STACK.
1520                    ;*
1521                    ;* CALLING SEQUENCE:    PASS    R0,R1,...
1522                    ;*
1523                    ;* COMMENTS:    ANY COMBINATION OF GPR ARGUMENTS MAY BE LISTED IN ANY ORDER.
1524                    ;*              FOR EXAMPLE, THE FOLLOWING ARE LEGAL:
1525                    ;*                      PASS    R1
1526                    ;*                      PASS    R4,R0,R2
1527                    ;*              THE GPRS LISTED AS ARGUMENTS WILL BE PASSED INTACT TO THE
1528                    ;*              CALLING ROUTINE, ALL OTHER GPRS WILL BE RESTORED.
1529                    ;*              THE SP MUST BE AT ITS ORIGINAL VALUE WHEN PASS IS CALLED.
1530                    ;*
1531                    ;*              THE MACRO CALL
1532                    ;*                      PASS    R0,R3
1533                    ;*              EXPANDS INTO THE FOLLOWING ASSEMBLY CODE:
1534                    ;*                      MOV     R0,R0SLOT(SP)           ;PUT R0 IN STACK SLOT.
1535                    ;*                      MOV     R3,R3SLOT(SP)           ;PUT R3 IN STACK SLOT.
1536                    ;*                      JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
1537                    ;*              IN THIS EXAMPLE GPRS R1, R2, R4, AND R5 WILL BE RESTORED TO
1538                    ;*              THEIR VALUES CONTAINED IN THE STACK FRAME AND R0 AND R3
1539                    ;*              WILL BE LEFT AT THEIR VALUES PRIOR TO THIS PASS CALL.
1540                    ;*
1541                    ;* SUBORDINATE ROUTINES CALLED: (PREGRT - LABEL WITHIN PREGO5, VALUE ON STACK.)
1542                    ;*****************************************************************************
1543
1544                            .MACRO  PASS    A,B,C,D,E,F
1545                            .IRP    X,<A,B,C,D,E,F>
1546                            .IF     NB,X
1547                            .LIST
1548                                    MOV     X,X'SLOT(SP)            ;PUT X IN STACK SLOT.
1549                            .NLIST
1550                            .ENDC
1551                            .ENDM
1552                            .LIST
1553                                    JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
1554                            .NLIST
1555                            .ENDM   PASS
```

```
                                .SBTTL GLOBAL SUBROUTINE                - PREGO5 -
                                ;**************************************************************************
                                ;*      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,PREGO5".)
                                ;*
                                ;* OUTPUTS:     REGISTERS R0 THROUGH R5 ARE SAVED ON THE STACK.
                                ;*
                                ;*CALLING SEQUENCE:     SAVE            ;MACRO EXPANSION CALLS PREGO5.
                                ;*                      [SUBROUTINE CODE]...
                                ;*                      PASS            ;MACRO EXPANSION RECALLS PREGO5.
                                ;*
                                ;*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.
                                ;**************************************************************************

003776                          PREGO5:                 ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
003776  010446                          MOV     R4,-(SP)        ;SAVE R4
004000  010346                          MOV     R3,-(SP)        ;SAVE R3
004002  010246                          MOV     R2,-(SP)        ;SAVE R2
004004  010146                          MOV     R1,-(SP)        ;SAVE R1
004006  010046                          MOV     R0,-(SP)        ;SAVE R0
004010  010546                          MOV     R5,-(SP)        ;PUSH RETURN PC ON TOP OF STACK
004012  016605  000014                  MOV     R5SLOT(SP),R5   ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS

004016  004736                          JSR     PC,@(SP)+       ;CALL THE SUBROUTINE AT THE RETURN ADDRESS
                                                                ;FROM THE PREGO5 CALL, PUTTING THE PRESENT
                                                                ;PC ON THE STACK AS A RETURN ADDRESS INTO
                                                                ;THIS (PREGO5) ROUTINE.

                                        ;+++
                                        ;THE FOLLOWING CODE IS EXECUTED WHEN THE CALLING ROUTINE DOES A
                                        ;"RETURN" [JSR  PC,@(SP)+] USING THE PC DEPOSITED ON THE STACK ABOVE.
                                        ;---

004020  012605                  PREGRT:: MOV    (SP)+,R5        ;PUT RETURN PC IN R5.
004022  012600                          MOV     (SP)+,R0        ;RESTORE R0.
004024  012601                          MOV     (SP)+,R1        ;RESTORE R1.
004026  012602                          MOV     (SP)+,R2        ;RESTORE R2.
004030  012603                          MOV     (SP)+,R3        ;RESTORE R3.
004032  012604                          MOV     (SP)+,R4        ;RESTORE R4.

004034  000205                          RTS     R5              ;RETURN TO THE SUBROUTINE WHICH CALLED PREGO5.
                                                                  ;RESTORING R5 IN THE PROCESS.
```

M3


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


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

                                ;
                                ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                                ;
                                        DEVTYP  <DHV-11>
004036                                                                  L$DVTYP::
004036                                                                          .ASCIZ  /DHV-11/
004036  104     110     126
004041  055     061     061
004044  000
                                                                                .EVEN


                                ; TEST DESCRIPTION
                                ;
                                        DESCRIPT        <DHV-11 FUNC TST PART1>
004046                                                                  L$DESC::
004046                                                                          .ASCIZ  /DHV-11 FUNC TST PAR
004046  104     110     126
T1/
004051  055     061     061
004054  040     106     125
004057  116     103     040
004062  124     123     124
004065  040     120     101
004070  122     124     061
004073  000
                                                                                .EVEN

                                .EVEN
```

```
                                ;*****************************************************************************
                                ;
                                ;               FVTA.FMT
                                ;
                                ;*****************************************************************************



                                ;
                                ; FORMAT STATEMENTS USED IN PRINT CALLS
                                ;
                                        ; ********** FORMAT STATEMENTS ***********
004074  045  116  045  MFUNIT:: .ASCIZ /%N%A TESTING UNIT :%D4%A(D)%N/
004077  101  040  124
004102  105  123  124
004105  111  116  107
004110  040  125  116
004113  111  124  040
004116  072  045  104
004121  064  045  101
004124  050  104  051
004127  045  116  000
004132  045  124  045  EF0503:: .ASCIZ /%T%N/
004135  116  000
004137  045  101  040  EF0505:: .ASCIZ /%A    %D5%A ILLEGAL INTERRUPTS RECEIVED.%N/
004142  040  040  040
004145  045  104  065
004150  045  101  040
004153  111  114  114
004156  105  107  101
004161  114  040  111
004164  116  124  105
004167  122  122  125
004172  120  124  123
004175  040  122  105
004200  103  105  111
004203  126  105  104
004206  056  045  116
004211  000
004212  045  116  045  EF1401:: .ASCIZ /%N%A ROM VERSION NUMBERS: PROC_1 = %D2%A(D)   PROC_2 = %D2%A(D)%N/
004215  101  040  122
004220  117  115  040
004223  126  105  122
004226  123  111  117
004231  116  040  116
004234  125  115  102
004237  105  122  123
004242  072  040  120
004245  122  117  103
004250  137  061  040
004253  075  040  045
004256  104  062  045
004261  101  050  104
004264  051  040  040
004267  040  120  122
004272  117  103  137
```

GLOBAL TEXT SECTION

```
        004275      062     040     075
        004300      040     045     104
        004303      062     045     101
        004306      050     104     051
        004311      045     116     000
1668    004314      045     124     045  EF1402:: .ASCIZ /%T%A ROM VERSION NUMBER %T%N/
        004317      101     040     122
        004322      117     115     040
        004325      126     105     122
        004330      123     111     117
        004333      116     040     116
        004336      125     115     102
        004341      105     122     040
        004344      045     124     045
        004347      116     000
1669    004351      045     101     040  EF1601:: .ASCIZ /%A  %T%A ABORTED %N/
        004354      040     045     124
        004357      045     101     040
        004362      101     102     117
        004365      122     124     105
        004370      104     040     045
        004373      116     000
1670    004375      045     101     040  EF1602:: .ASCIZ /%A    EXPECTED DATA: %06%A (O).%N/
        004400      040     040     040
        004403      105     130     120
        004406      105     103     124
        004411      105     104     040
        004414      104     101     124
        004417      101     072     040
        004422      045     117     066
        004425      045     101     040
        004430      050     117     051
        004433      056     045     116
        004436      000
1671    004437      045     101     040  EF1603:: .ASCIZ /%A    ACTUAL DATA:   %06%A (O).%N/
        004442      040     040     040
        004445      101     103     124
        004450      125     101     114
        004453      040     104     101
        004456      124     101     072
        004461      040     040     040
        004464      045     117     066
        004467      045     101     040
        004472      050     117     051
        004475      056     045     116
        004500      000
1672    004501      045     101     040  EF1604:: .ASCIZ /%A  BAD BIT(S) IN DEVICE %T%A REGISTER FOR LINE %D2%A (D).%N/
        004504      040     102     101
        004507      104     040     102
        004512      111     124     050
        004515      123     051     040
        004520      111     116     040
        004523      104     105     126
        004526      111     103     105
        004531      040     045     124
        004534      045     101     040
        004537      122     105     107
```

```
        004542  111  123  124
        004545  105  122  040
        004550  106  117  122
        004553  040  114  111
        004556  116  105  040
        004561  045  104  062
        004564  045  101  040
        004567  050  104  051
        004572  056  045  116
        004575  000
1673    004576  045  101  040  EF3001:: .ASCIZ /%A    EXPECTED OR CORRECT VALUE: %O3%N/
        004601  040  040  040
        004604  105  130  120
        004607  105  103  124
        004612  105  104  040
        004615  117  122  040
        004620  103  117  122
        004623  122  105  103
        004626  124  040  126
        004631  101  114  125
        004634  105  072  040
        004637  045  117  063
        004642  045  116  000
1674    004645  045  101  040  EF3002:: .ASCIZ /%A    ACTUAL OR MEASURED  VALUE: %O3%N/
        004650  040  040  040
        004653  101  103  124
        004656  125  101  114
        004661  040  117  122
        004664  040  115  105
        004667  101  123  125
        004672  122  105  104
        004675  040  040  126
        004700  101  114  125
        004703  105  072  040
        004706  045  117  063
        004711  045  116  000
1675    004714  045  101  040  EF9001:: .ASCIZ /%A  UNEXPECTED %T%A FOUND IN RECEIVE CHAR FIFO:%N/
        004717  040  125  116
        004722  105  130  120
        004725  105  103  124
        004730  105  104  040
        004733  045  124  045
        004736  101  040  106
        004741  117  125  116
        004744  104  040  111
        004747  116  040  122
        004752  105  103  105
        004755  111  126  105
        004760  040  103  110
        004763  101  122  040
        004766  106  111  106
        004771  117  072  045
        004774  116  000
1676    004776  045  101  040  EF9002:: .ASCIZ /%A    CODE IS ASSOCIATED WITH LINE: %D2%A(D)%N/
        005001  040  040  040
        005004  103  117  104
        005007  105  040  111
```

GLOBAL TEXT SECTION

```
        005012      123     040     101
        005015      123     123     117
        005020      103     111     101
        005023      124     105     104
        005026      040     127     111
        005031      124     110     040
        005034      114     111     116
        005037      105     072     040
        005042      045     104     062
        005045      045     101     050
        005050      104     051     045
        005053      116     000
1677    005055      045     101     040  EF9003:: .ASCIZ /%A      CODE IS: %O3%A(O)%N/
        005060      040     040     040
        005063      040     040     103
        005066      117     104     105
        005071      040     111     123
        005074      072     040     045
        005077      117     063     045
        005102      101     050     117
        005105      051     045     116
        005110      000
1678    005111      045     101     040  EF9004:: .ASCIZ /%A    %T%A VALUE: %O3%A(O)%N/
        005114      040     040     040
        005117      045     124     045
        005122      101     040     126
        005125      101     114     125
        005130      105     072     040
        005133      045     117     063
        005136      045     101     050
        005141      117     051     045
        005144      116     000
1679    005146      045     101     040  EF9005:: .ASCIZ /%A    %T%A VALUE: NONE%N/
        005151      040     040     040
        005154      045     124     045
        005157      101     040     126
        005162      101     114     125
        005165      105     072     040
        005170      116     117     116
        005173      105     045     116
        005176      000
1680    005177      045     101     040  EF9006:: .ASCIZ /%A  %T%A %D2%A(D)%N/
        005202      040     045     124
        005205      045     101     040
        005210      045     104     062
        005213      045     101     050
        005216      104     051     045
        005221      116     000
1681    005223      045     101     040  EF9010:: .ASCIZ /%A    NUMBER OF ERRORS DETECTED ON LINE %D2%A(D) IS %D5%A(D)%N/
        005226      040     040     040
        005231      116     125     115
        005234      102     105     122
        005237      040     117     106
        005242      040     105     122
        005245      122     117     122
        005250      123     040     104
        005253      105     124     105
```

```
        005256      103     124     105
        005261      104     040     117
        005264      116     040     114
        005267      111     116     105
        005272      040     045     104
        005275      062     045     101
        005300      050     104     051
        005303      040     111     123
        005306      040     045     104
        005311      065     045     101
        005314      050     104     051
        005317      045     116     000
1682    005322      045     101     040  EF9016:: .ASCIZ /%A  UNEXPECTED %T%A FOR LINE %D2%A(D) IN FIFO AFTER RESET:%N/
        005325      040     125     116
        005330      105     130     120
        005333      105     103     124
        005336      105     104     040
        005341      045     124     045
        005344      101     040     106
        005347      117     122     040
        005352      114     111     116
        005355      105     040     045
        005360      104     062     045
        005363      101     050     104
        005366      051     040     111
        005371      116     040     106
        005374      111     106     117
        005377      040     101     106
        005402      124     105     122
        005405      040     122     105
        005410      123     105     124
        005413      072     045     116
        005416      000
1683    005417      045     101     040  EF9017:: .ASCIZ /%A    %T%A (WITH ERROR FLAGS) IS %O6%A(0)%N/
        005422      040     040     040
        005425      045     124     045
        005430      101     040     050
        005433      127     111     124
        005436      110     040     105
        005441      122     122     117
        005444      122     040     106
        005447      114     101     107
        005452      123     051     040
        005455      111     123     040
        005460      045     117     066
        005463      045     101     050
        005466      117     051     045
        005471      116     000
1684    005473      045     101     040  EF9018:: .ASCII /%A  %T%A IN SELFTEST CODE FIFO SLOT FOR LINE %D2/
        005476      040     045     124
        005501      045     101     040
        005504      111     116     040
        005507      123     105     114
        005512      106     124     105
        005515      123     124     040
        005520      103     117     104
        005523      105     040     106
```

```
        005526      111     106     117
        005531      040     123     114
        005534      117     124     040
        005537      106     117     122
        005542      040     114     111
        005545      116     105     040
        005550      045     104     062
1685    005553      045     101     050             .ASCIZ /%A(D) AFTER RESET.%N/
        005556      104     051     040
        005561      101     106     124
        005564      105     122     040
        005567      122     105     123
        005572      105     124     056
        005575      045     116     000
1686    005600      045     101     040     EF9019:: .ASCIZ /%A  %T%A %06%A(0)%N/
        005603      040     045     124
        005606      045     101     040
        005611      045     117     066
        005614      045     101     050
        005617      117     051     045
        005622      116     000
1687    005624      045     101     040     EF9301:: .ASCIZ /%A %T%D2%A(D),  BMP CODE REPORTED :%03%A(0)%N/
        005627      045     124     045
        005632      104     062     045
        005635      101     050     104
        005640      051     054     040
        005643      040     102     115
        005646      120     040     103
        005651      117     104     105
        005654      040     122     105
        005657      120     117     122
        005662      124     105     104
        005665      040     072     045
        005670      117     063     045
        005673      101     050     117
        005676      051     045     116
        005701      000
1688    005702      045     101     040     EF9302:: .ASCIZ /%A  OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)%N/
        005705      040     117     126
        005710      105     122     106
        005713      114     117     127
        005716      040     117     103
        005721      103     125     122
        005724      122     105     104
        005727      040     050     115
        005732      117     122     105
        005735      040     124     110
        005740      101     116     040
        005743      063     061     040
        005746      102     115     120
        005751      040     103     117
        005754      104     105     123
        005757      040     106     117
        005762      125     116     104
        005765      040     111     116
        005770      040     121     125
        005773      105     125     105
```

GLOBAL TEXT SECTION

```
        005776      051     045     116
        006001      000
  1689
  1690                                          .EVEN
  1701
  1702
```

```
1711
1712                            ;**********************************************************************************************
1713                            ;
1714                            ;               FVTA.MSG
1715                            ;
1716                            ;**********************************************************************************************
**
1717
1718
1719                            .SBTTL  GLOBAL MESSAGE AREA
1720                                    ;********** ERROR MESSAGES ************
1721 006002  103  123  122  DR00MG:: .ASCIZ /CSR/
     006005  000
1722 006006  122  102  125  DR02MG:: .ASCIZ /RBUF/
     006011  106  000
1723 006013  114  120  122  DR04MG:: .ASCIZ /LPR/
     006016  000
1724 006017  123  124  101  DR06MG:: .ASCIZ /STAT/
     006022  124  000
1725 006024  114  116  103  DR10MG:: .ASCIZ /LNCTRL/
     006027  124  122  114
     006032  000
1726 006033  124  102  125  DR12MG:: .ASCIZ /TBUFFAD1/
     006036  106  106  101
     006041  104  061  000
1727 006044  124  102  125  DR14MG:: .ASCIZ /TBUFFAD2/
     006047  106  106  101
     006052  104  062  000
1728 006055  124  102  125  DR16MG:: .ASCIZ /TBUFFCT/
     006060  106  106  103
     006063  124  000
1729 006065  104  105  126  EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
     006070  111  103  105
     006073  040  122  105
     006076  107  111  123
     006101  124  105  122
     006104  040  101  103
     006107  103  105  123
     006112  123  040  105
     006115  122  122  117
     006120  122  123  000
1730 006123  115  101  123  EM0201:: .ASCIZ /MASTER RESET (PERFORM SELFTEST) TEST /
     006126  124  105  122
     006131  040  122  105
     006134  123  105  124
     006137  040  050  120
     006142  105  122  106
     006145  117  122  115
     006150  040  123  105
     006153  114  106  124
     006156  105  123  124
     006161  051  040  124
     006164  105  123  124
     006167  040  000
1731 006171  040  040  115  EM0202:: .ASCIZ /  MASTER RESET BIT DID NOT CLEAR AFTER BOARD RESET./
     006174  101  123  124
     006177  105  122  040
     006202  122  105  123
```

```
            006205  105  124  040
            006210  102  111  124
            006213  040  104  111
            006216  104  040  116
            006221  117  124  040
            006224  103  114  105
            006227  101  122  040
            006232  101  106  124
            006235  105  122  040
            006240  102  117  101
            006243  122  104  040
            006246  122  105  123
            006251  105  124  056
            006254  000
1732        006255  040  040  040          .ASCIZ /    WAITED 5 SECONDS.  BIT DEFECTIVE OR FIRMWARE HUNG./
            006260  040  127  101
            006263  111  124  105
            006266  104  040  065
            006271  040  123  105
            006274  103  117  116
            006277  104  123  056
            006302  040  040  102
            006305  111  124  040
            006310  104  105  106
            006313  105  103  124
            006316  111  126  105
            006321  040  117  122
            006324  040  106  111
            006327  122  115  127
            006332  101  122  105
            006335  040  110  125
            006340  116  107  056
            006343  000
1733        006344  040  040  115  EM0203:: .ASCIZ /  MASTER RESET BIT CLEAR IMMEDIATELY AFTER BOARD RESET./
            006347  101  123  124
            006352  105  122  040
            006355  122  105  123
            006360  105  124  040
            006363  102  111  124
            006366  040  103  114
            006371  105  101  122
            006374  040  111  115
            006377  115  105  104
            006402  111  101  124
            006405  105  114  131
            006410  040  101  106
            006413  124  105  122
            006416  040  102  117
            006421  101  122  104
            006424  040  122  105
            006427  123  105  124
            006432  056  000
1734        006434  040  040  040          .ASCIZ /    BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
            006437  040  102  111
            006442  124  040  104
            006445  105  106  105
            006450  103  124  111
```

```
        006453  126  105  040
        006456  117  122  040
        006461  102  117  101
        006464  122  104  040
        006467  106  111  122
        006472  115  127  101
        006475  122  105  040
        006500  105  122  122
        006503  117  122  056
        006506  000
1735    006507  040  040  115  EM0204:: .ASCIZ \  MR BIT WENT CLEAR WITHIN 1/2 SECOND OF BOARD RESET.\
        006512  122  040  102
        006515  111  124  040
        006520  127  105  116
        006523  124  040  103
        006526  114  105  101
        006531  122  040  127
        006534  111  124  110
        006537  111  116  040
        006542  061  057  062
        006545  040  123  105
        006550  103  117  116
        006553  104  040  117
        006556  106  040  102
        006561  117  101  122
        006564  104  040  122
        006567  105  123  105
        006572  124  056  000
1736    006575  040  040  040          .ASCIZ /    BIT DEFECTIVE OR SELFTEST WAS (INCORRECTLY) SKIPPED./
        006600  040  102  111
        006603  124  040  104
        006606  105  106  105
        006611  103  124  111
        006614  126  105  040
        006617  117  122  040
        006622  123  105  114
        006625  106  124  105
        006630  123  124  040
        006633  127  101  123
        006636  040  050  111
        006641  116  103  117
        006644  122  122  105
        006647  103  124  114
        006652  131  051  040
        006655  123  113  111
        006660  120  120  105
        006663  104  056  000
1737    006666  115  101  123  EM0301:: .ASCIZ /MASTER RESET (SKIP SELFTEST) TEST /
        006671  124  105  122
        006674  040  122  105
        006677  123  105  124
        006702  040  050  123
        006705  113  111  120
        006710  040  123  105
        006713  114  106  124
        006716  105  123  124
        006721  051  040  124
```

```
        006724  105  123  124
        006727  040  000
1738    006731  040  040  115  EM0302:: .ASCIZ /  MR BIT CLR WITHIN 10 MILISECOND AFTER BOARD RESET./
        006734  122  040  102
        006737  111  124  040
        006742  103  114  122
        006745  040  127  111
        006750  124  110  111
        006753  116  040  061
        006756  060  040  115
        006761  111  114  111
        006764  123  105  103
        006767  117  116  104
        006772  040  101  106
        006775  124  105  122
        007000  040  102  117
        007003  101  122  104
        007006  040  122  105
        007011  123  105  124
        007014  056  000
1739    007016  040  040  040          .ASCIZ /    BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
        007021  040  102  111
        007024  124  040  104
        007027  105  106  105
        007032  103  124  111
        007035  126  105  040
        007040  117  122  040
        007043  102  117  101
        007046  122  104  040
        007051  106  111  122
        007054  115  127  101
        007057  122  105  040
        007062  105  122  122
        007065  117  122  056
        007070  000
1740    007071  040  040  115  EM0303:: .ASCIZ \  MR BIT WENT CLEAR 1/5 TO 5 SECONDS AFTER RESET.\
        007074  122  040  102
        007077  111  124  040
        007102  127  105  116
        007105  124  040  103
        007110  114  105  101
        007113  122  040  061
        007116  057  065  040
        007121  124  117  040
        007124  065  040  123
        007127  105  103  117
        007132  116  104  123
        007135  040  101  106
        007140  124  105  122
        007143  040  122  105
        007146  123  105  124
        007151  056  000
1741    007153  040  040  040          .ASCIZ /    SELFTEST DID NOT GET SKIPPED (SHOULD HAVE BEEN SKIPPED)./
        007156  040  123  105
        007161  114  106  124
        007164  105  123  124
        007167  040  104  111
```

```
        007172      104     040     116
        007175      117     124     040
        007200      107     105     124
        007203      040     123     113
        007206      111     120     120
        007211      105     104     040
        007214      050     123     110
        007217      117     125     114
        007222      104     040     110
        007225      101     126     105
        007230      040     102     105
        007233      105     116     040
        007236      123     113     111
        007241      120     120     105
        007244      104     051     056
        007247      000
1742    007250      122     102     125  EM0401:: .ASCIZ /RBUF REGISTER RX CHARACTER FIELD TEST /
        007253      106     040     122
        007256      105     107     111
        007261      123     124     105
        007264      122     040     122
        007267      130     040     103
        007272      110     101     122
        007275      101     103     124
        007300      105     122     040
        007303      106     111     105
        007306      114     104     040
        007311      124     105     123
        007314      124     040     000
1743    007317      040     040     111  EM0402:: .ASCIZ /  IMPROPER CODE FOUND IN RX FIFO AFTER DUT RESET./
        007322      115     120     122
        007325      117     120     105
        007330      122     040     103
        007333      117     104     105
        007336      040     106     117
        007341      125     116     104
        007344      040     111     116
        007347      040     122     130
        007352      040     106     111
        007355      106     117     040
        007360      101     106     124
        007363      105     122     040
        007366      104     125     124
        007371      040     122     105
        007374      123     105     124
        007377      056     000
1744    007401      040     040     040          .ASCIZ /    EXPECTED: SELFTEST CODE,   ACTUAL: IMPROPER CODE./
        007404      040     105     130
        007407      120     105     103
        007412      124     105     104
        007415      072     040     123
        007420      105     114     106
        007423      124     105     123
        007426      124     040     103
        007431      117     104     105
        007434      054     040     040
        007437      040     101     103
```

```
                007442      124     125     101
                007445      114     072     040
                007450      111     115     120
                007453      122     117     120
                007456      105     122     040
                007461      103     117     104
                007464      105     056     000
1745            007467      122     102     125     EM0501:: .ASCIZ /RBUF REGISTER ERROR FLAGS FIELD TEST /
                007472      106     040     122
                007475      105     107     111
                007500      123     124     105
                007503      122     040     105
                007506      122     122     117
                007511      122     040     106
                007514      114     101     107
                007517      123     040     106
                007522      111     105     114
                007525      104     040     124
                007530      105     123     124
                007533      040     000
1746            007535      040     040     122     EM0502:: .ASCIZ /  RX ERROR FLAG(S) FOUND CLEAR ON SELFTEST CODE./
                007540      130     040     105
                007543      122     122     117
                007546      122     040     106
                007551      114     101     107
                007554      050     123     051
                007557      040     106     117
                007562      125     116     104
                007565      040     103     114
                007570      105     101     122
                007573      040     117     116
                007576      040     123     105
                007601      114     106     124
                007604      105     123     124
                007607      040     103     117
                007612      104     105     056
                007615      000
1747            007616      040     040     040             .ASCIZ /    EXPECTED: ALL ERROR FLAGS SET,  ACTUAL: FLAG(S) CLEAR./
                007621      040     105     130
                007624      120     105     103
                007627      124     105     104
                007632      072     040     101
                007635      114     114     040
                007640      105     122     122
                007643      117     122     040
                007646      106     114     101
                007651      107     123     040
                007654      123     105     124
                007657      054     040     040
                007662      101     103     124
                007665      125     101     114
                007670      072     040     106
                007673      114     101     107
                007676      050     123     051
                007701      040     103     114
                007704      105     101     122
                007707      056     000
```

```
1748 007711  040  040  122  EM0525:: .ASCIZ /  RX INTERRUPT(S) RECEIVED WITH RX INTERRUPTS DISABLED./
     007714  130  040  111
     007717  116  124  105
     007722  122  122  125
     007725  120  124  050
     007730  123  051  040
     007733  122  105  103
     007736  105  111  126
     007741  105  104  040
     007744  127  111  124
     007747  110  040  122
     007752  130  040  111
     007755  116  124  105
     007760  122  122  125
     007763  120  124  123
     007766  040  104  111
     007771  123  101  102
     007774  114  105  104
     007777  056  000
1749 010001  040  040  124  EM0526:: .ASCIZ /  TX INTERRUPT(S) RECEIVED WITH TX INTERRUPTS DISABLED./
     010004  130  040  111
     010007  116  124  105
     010012  122  122  125
     010015  120  124  050
     010020  123  051  040
     010023  122  105  103
     010026  105  111  126
     010031  105  104  040
     010034  127  111  124
     010037  110  040  124
     010042  130  040  111
     010045  116  124  105
     010050  122  122  125
     010053  120  124  123
     010056  040  104  111
     010061  123  101  102
     010064  114  105  104
     010067  056  000
1750 010071  103  123  122  EM0601:: .ASCIZ /CSR RX.DATA.AVAIL BIT TEST /
     010074  040  122  130
     010077  056  104  101
     010102  124  101  056
     010105  101  126  101
     010110  111  114  040
     010113  102  111  124
     010116  040  124  105
     010121  123  124  040
     010124  000
1751 010125  040  040  122  EM0602:: .ASCIZ /  RX.DATA.AVAIL BIT FOUND CLEAR AFTER RESET COMPLETION./
     010130  130  056  104
     010133  101  124  101
     010136  056  101  126
     010141  101  111  114
     010144  040  102  111
     010147  124  040  106
     010152  117  125  116
     010155  104  040  103
```

```
        010160  114  105  101
        010163  122  040  101
        010166  106  124  105
        010171  122  040  122
        010174  105  123  105
        010177  124  040  103
        010202  117  115  120
        010205  114  105  124
        010210  111  117  116
        010213  056  000
1752    010215  040  040  040          .ASCIZ /    EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
        010220  040  105  130
        010223  120  105  103
        010226  124  105  104
        010231  040  102  111
        010234  124  040  124
        010237  117  040  102
        010242  105  040  123
        010245  105  124  040
        010250  106  122  117
        010253  115  040  123
        010256  105  114  106
        010261  124  105  123
        010264  124  040  103
        010267  117  104  105
        010272  123  040  111
        010275  116  040  106
        010300  111  106  117
        010303  056  000
1753    010305  040  040  122  EM0603:: .ASCIZ /  RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO./
        010310  130  056  104
        010313  101  124  101
        010316  056  101  126
        010321  101  111  114
        010324  040  102  111
        010327  124  040  103
        010332  117  125  114
        010335  104  040  116
        010340  117  124  040
        010343  102  105  040
        010346  103  114  105
        010351  101  122  105
        010354  104  040  102
        010357  131  040  120
        010362  125  122  107
        010365  111  116  107
        010370  040  106  111
        010373  106  117  056
        010376  000
1754    010377  040  040  040          .ASCIZ /    600 CHARS READ FROM FIFO WITHOUT R.D.A BIT CLEARING./
        010402  040  066  060
        010405  060  040  103
        010410  110  101  122
        010413  123  040  122
        010416  105  101  104
        010421  040  106  122
        010424  117  115  040
```

```
        010427  106  111  106
        010432  117  040  127
        010435  111  124  110
        010440  117  125  124
        010443  040  122  056
        010446  104  056  101
        010451  040  102  111
        010454  124  040  103
        010457  114  105  101
        010462  122  111  116
        010465  107  056  000
1755    010470  122  102  125  EM0701:: .ASCIZ /RBUF RX.DATA.VALID BIT TEST /
        010473  106  040  122
        010476  130  056  104
        010501  101  124  101
        010504  056  126  101
        010507  114  111  104
        010512  040  102  111
        010515  124  040  124
        010520  105  123  124
        010523  040  000
1756    010525  040  040  122  EM0702:: .ASCIZ /  RX.DATA.VALID BIT FOUND CLEAR AFTER RESET COMPLETION./
        010530  130  056  104
        010533  101  124  101
        010536  056  126  101
        010541  114  111  104
        010544  040  102  111
        010547  124  040  106
        010552  117  125  116
        010555  104  040  103
        010560  114  105  101
        010563  122  040  101
        010566  106  124  105
        010571  122  040  122
        010574  105  123  105
        010577  124  040  103
        010602  117  115  120
        010605  114  105  124
        010610  111  117  116
        010613  056  000
1757    010615  040  040  040          .ASCIZ /    EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
        010620  040  105  130
        010623  120  105  103
        010626  124  105  104
        010631  040  102  111
        010634  124  040  124
        010637  117  040  102
        010642  105  040  123
        010645  105  124  040
        010650  106  122  117
        010653  115  040  123
        010656  105  114  106
        010661  124  105  123
        010664  124  040  103
        010667  117  104  105
        010672  123  040  111
        010675  116  040  106
```

```
        010700  111  106  117
        010703  056  000
1758    010705  040  040  122  EM0703:: .ASCIZ /  RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO./
        010710  130  056  104
        010713  101  124  101
        010716  056  126  101
        010721  114  111  104
        010724  040  102  111
        010727  124  040  103
        010732  117  125  114
        010735  104  040  116
        010740  117  124  040
        010743  102  105  040
        010746  103  114  105
        010751  101  122  105
        010754  104  040  102
        010757  131  040  120
        010762  125  122  107
        010765  111  116  107
        010770  040  106  111
        010773  106  117  056
        010776  000
1759    010777  040  040  040          .ASCIZ /    600 CHARS READ FROM FIFO WITHOUT R.D.V BIT CLEARING./
        011002  040  066  060
        011005  060  040  103
        011010  110  101  122
        011013  123  040  122
        011016  105  101  104
        011021  040  106  122
        011024  117  115  040
        011027  106  111  106
        011032  117  040  127
        011035  111  124  110
        011040  117  125  124
        011043  040  122  056
        011046  104  056  126
        011051  040  102  111
        011054  124  040  103
        011057  114  105  101
        011062  122  111  116
        011065  107  056  000
1760    011070  122  102  125  EM0801:: .ASCIZ /RBUF RX.LINE.NUMBER FIELD TEST /
        011073  106  040  122
        011076  130  056  114
        011101  111  116  105
        011104  056  116  125
        011107  115  102  105
        011112  122  040  106
        011115  111  105  114
        011120  104  040  124
        011123  105  123  124
        011126  040  000
1761    011130  040  040  114  EM0802:: .ASCIZ /  LINE NUMBER IS WRONG ON A SELFTEST CODE./
        011133  111  116  105
        011136  040  116  125
        011141  115  102  105
        011144  122  040  111
```

```
        011147  123  040  127
        011152  122  117  116
        011155  107  040  117
        011160  116  040  101
        011163  040  123  105
        011166  114  106  124
        011171  105  123  124
        011174  040  103  117
        011177  104  105  056
        011202  000
1762    011203  103  110  105  EM0901:: .ASCIZ /CHECK FOR BMP_CODES TEST/
        011206  103  113  040
        011211  106  117  122
        011214  040  102  115
        011217  120  137  103
        011222  117  104  105
        011225  123  040  124
        011230  105  123  124
        011233  000
1763    011234  125  116  105  EM0902:: .ASCIZ /UNEXPECTED BMP CODES FOUND./
        011237  130  120  105
        011242  103  124  105
        011245  104  040  102
        011250  115  120  040
        011253  103  117  104
        011256  105  123  040
        011261  106  117  125
        011264  116  104  056
        011267  000
1764    011270  104  111  101  EM1001:: .ASCIZ /DIAGNOSTIC FAIL (SKP SELFTEST) TEST/
        011273  107  116  117
        011276  123  124  111
        011301  103  040  106
        011304  101  111  114
        011307  040  050  123
        011312  113  120  040
        011315  123  105  114
        011320  106  124  105
        011323  123  124  051
        011326  040  124  105
        011331  123  124  000
1765    011334  040  040  123  EM1002:: .ASCIZ /  SKIP SELF-TEST TOOK TOO LONG TO COMPLETE, > 50 MS./
        011337  113  111  120
        011342  040  123  105
        011345  114  106  055
        011350  124  105  123
        011353  124  040  124
        011356  117  117  113
        011361  040  124  117
        011364  117  040  114
        011367  117  116  107
        011372  040  124  117
        011375  040  103  117
        011400  115  120  114
        011403  105  124  105
        011406  054  040  076
        011411  040  065  060
```

```
        011414      040      115      123
        011417      056      000
1766    011421      040      040      123  EM1003:: .ASCIZ /  SKIP SELF-TEST COMPLETED TOO SOON, < 10 MS./
        011424      113      111      120
        011427      040      123      105
        011432      114      106      055
        011435      124      105      123
        011440      124      040      103
        011443      117      115      120
        011446      114      105      124
        011451      105      104      040
        011454      124      117      117
        011457      040      123      117
        011462      117      116      054
        011465      040      074      040
        011470      061      060      040
        011473      115      123      056
        011476      000
1767    011477      123      113      111  EM1101:: .ASCIZ /SKIP SELF-TEST TEST/
        011502      120      040      123
        011505      105      114      106
        011510      055      124      105
        011513      123      124      040
        011516      124      105      123
        011521      124      000
1768    011523      123      105      114  EM1201:: .ASCIZ /SELF-TEST TEST/
        011526      106      055      124
        011531      105      123      124
        011534      040      124      105
        011537      123      124      000
1769    011542      040      040      123  EM1202:: .ASCIZ /  SELF-TEST TOOK TOO LONG TO COMPLETE, > 3 SECONDS./
        011545      105      114      106
        011550      055      124      105
        011553      123      124      040
        011556      124      117      117
        011561      113      040      124
        011564      117      117      040
        011567      114      117      116
        011572      107      040      124
        011575      117      040      103
        011600      117      115      120
        011603      114      105      124
        011606      105      054      040
        011611      076      040      063
        011614      040      123      105
        011617      103      117      116
        011622      104      123      056
        011625      000
1770    011626      040      040      123  EM1203:: .ASCIZ \  SELF-TEST COMPLETED TOO SOON, < 1/2 SECOND.\
        011631      105      114      106
        011634      055      124      105
        011637      123      124      040
        011642      103      117      115
        011645      120      114      105
        011650      124      105      104
        011653      040      124      117
        011656      117      040      123
```

```
        011661      117     117     116
        011664      054     040     074
        011667      040     061     057
        011672      062     040     123
        011675      105     103     117
        011700      116     104     056
        011703      000
1771    011704      040     040     123  EM1204:: .ASCIZ /  SELF-TEST DID NOT EXECUTE/
        011707      105     114     106
        011712      055     124     105
        011715      123     124     040
        011720      104     111     104
        011723      040     116     117
        011726      124     040     105
        011731      130     105     103
        011734      125     124     105
        011737      000
1772    011740      040     040     104  EM1205:: .ASCIZ /  DIAG_FAIL BIT BAD/
        011743      111     101     107
        011746      137     106     101
        011751      111     114     040
        011754      102     111     124
        011757      040     102     101
        011762      104     000
1773    011764      106     101     111  EM1301:: .ASCIZ /FAIL SELF-TEST TEST/
        011767      114     040     123
        011772      105     114     106
        011775      055     124     105
        012000      123     124     040
        012003      124     105     123
        012006      124     000
1774    012010      040     123     105  EM1302:: .ASCIZ / SELF-TEST ERROR REPORTING BAD/
        012013      114     106     055
        012016      124     105     123
        012021      124     040     105
        012024      122     122     117
        012027      122     040     122
        012032      105     120     117
        012035      122     124     111
        012040      116     107     040
        012043      102     101     104
        012046      000
1775    012047      122     117     115  EM1401:: .ASCIZ /ROM VERSION_NUMBER TEST/
        012052      040     126     105
        012055      122     123     111
        012060      117     116     137
        012063      116     125     115
        012066      102     105     122
        012071      040     124     105
        012074      123     124     000
1776    012077      040     040     106  EM1402:: .ASCIZ /  FIFO EMPTY, ONE OR MORE ROM VERSION_NUMBERS MISSING/
        012102      111     106     117
        012105      040     105     115
        012110      120     124     131
        012113      054     040     117
        012116      116     105     040
        012121      117     122     040
```

```
        012124  115     117     122
        012127  105     040     122
        012132  117     115     040
        012135  126     105     122
        012140  123     111     117
        012143  116     137     116
        012146  125     115     102
        012151  105     122     123
        012154  040     115     111
        012157  123     123     111
        012162  116     107     000
1777    012165  040     040     122     EM1403:: .ASCIZ /  ROM VERSION_NUMBER FOUND OUT OF SEQUENCE/
        012170  117     115     040
        012173  126     105     122
        012176  123     111     117
        012201  116     137     116
        012204  125     115     102
        012207  105     122     040
        012212  106     117     125
        012215  116     104     040
        012220  117     125     124
        012223  040     117     106
        012226  040     123     105
        012231  121     125     105
        012234  116     103     105
        012237  000
1778    012240  040     040     117     EM1404:: .ASCIZ /  ONE OR MORE ROM VERSION_NUMBERS MISSING/
        012243  116     105     040
        012246  117     122     040
        012251  115     117     122
        012254  105     040     122
        012257  117     115     040
        012262  126     105     122
        012265  123     111     117
        012270  116     137     116
        012273  125     115     102
        012276  105     122     123
        012301  040     115     111
        012304  123     123     111
        012307  116     107     000
1779    012312  040     040     040     EM1405:: .ASCIZ /    PROC_1/
        012315  040     120     122
        012320  117     103     137
        012323  061     000
1780    012325  040     040     040     EM1406:: .ASCIZ /    PROC_2/
        012330  040     120     122
        012333  117     103     137
        012336  062     000
1781    012340  116     117     124     EM1407:: .ASCIZ /NOT FOUND/
        012343  040     106     117
        012346  125     116     104
        012351  000
1782    012352  106     117     125     EM1408:: .ASCIZ /FOUND/
        012355  116     104     000
1783    012360  124     111     115     EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
        012363  105     117     125
        012366  124     040     117
```

```
        012371      103     103     125
        012374      122     122     105
        012377      104     040     127
        012402      101     111     124
        012405      111     116     107
        012410      040     106     117
        012413      122     040     115
        012416      101     123     124
        012421      105     122     040
        012424      122     105     123
        012427      105     124     040
        012432      124     117     040
        012435      103     114     105
        012440      101     122     000
1784    012443      104     105     126  EM1604:: .ASCIZ \DEVICE REGISTER WORD READ/WRITE TEST \
        012446      111     103     105
        012451      040     122     105
        012454      107     111     123
        012457      124     105     122
        012462      040     127     117
        012465      122     104     040
        012470      122     105     101
        012473      104     057     127
        012476      122     111     124
        012501      105     040     124
        012504      105     123     124
        012507      040     000
1785    012511      104     105     126  EM1701:: .ASCIZ \DEVICE REGISTER WORD READ/MODIFY/WRITE TEST \
        012514      111     103     105
        012517      040     122     105
        012522      107     111     123
        012525      124     105     122
        012530      040     127     117
        012533      122     104     040
        012536      122     105     101
        012541      104     057     115
        012544      117     104     111
        012547      106     131     057
        012552      127     122     111
        012555      124     105     040
        012560      124     105     123
        012563      124     040     000
1786    012566      104     105     126  EM1801:: .ASCIZ \DEVICE REGISTER BYTE READ/WRITE TEST \
        012571      111     103     105
        012574      040     122     105
        012577      107     111     123
        012602      124     105     122
        012605      040     102     131
        012610      124     105     040
        012613      122     105     101
        012616      104     057     127
        012621      122     111     124
        012624      105     040     124
        012627      105     123     124
        012632      040     000
1787    012634      104     105     126  EM1901:: .ASCIZ \DEVICE REGISTER BYTE READ/MODIFY/WRITE TEST \
        012637      111     103     105
```

```
        012642  040     122     105
        012645  107     111     123
        012650  124     105     122
        012653  040     102     131
        012656  124     105     040
        012661  122     105     101
        012664  104     057     115
        012667  117     104     111
        012672  106     131     057
        012675  127     122     111
        012700  124     105     040
        012703  124     105     123
        012706  124     040     000
1788    012711  104     105     126  EM2001:: .ASCIZ /DEVICE STAT REGISTER ID BIT TEST /
        012714  111     103     105
        012717  040     123     124
        012722  101     124     040
        012725  122     105     107
        012730  111     123     124
        012733  105     122     040
        012736  111     104     040
        012741  102     111     124
        012744  040     124     105
        012747  123     124     040
        012752  000
1789    012753  111     104     040  EM2002:: .ASCIZ /ID BIT BAD.  EXPECTED: CLEAR, ACTUAL: SET./
        012756  102     111     124
        012761  040     102     101
        012764  104     056     040
        012767  040     105     130
        012772  120     105     103
        012775  124     105     104
        013000  072     040     103
        013003  114     105     101
        013006  122     054     040
        013011  101     103     124
        013014  125     101     114
        013017  072     040     123
        013022  105     124     056
        013025  000
1790    013026  116     117     040  EM2101:: .ASCIZ \NO TX_DATA_VALID/NO TX_ACTION TEST\
        013031  124     130     137
        013034  104     101     124
        013037  101     137     126
        013042  101     114     111
        013045  104     057     116
        013050  117     040     124
        013053  130     137     101
        013056  103     124     111
        013061  117     116     040
        013064  124     105     123
        013067  124     000
1791    013071  040     040     124  EM2102:: .ASCIZ /  TX_ACTION FOUND AFTER INVALID DATA WORD WRITTEN TO LINE: /
        013074  130     137     101
        013077  103     124     111
        013102  117     116     040
        013105  106     117     125
```

```
        013110      116      104      040
        013113      101      106      124
        013116      105      122      040
        013121      111      116      126
        013124      101      114      111
        013127      104      040      104
        013132      101      124      101
        013135      040      127      117
        013140      122      104      040
        013143      127      122      111
        013146      124      124      105
        013151      116      040      124
        013154      117      040      114
        013157      111      116      105
        013162      072      040      000
1792    013165      124      130      137  EM2201:: .ASCIZ \TX_DATA_VALID/TX_ACTION TEST\
        013170      104      101      124
        013173      101      137      126
        013176      101      114      111
        013201      104      057      124
        013204      130      137      101
        013207      103      124      111
        013212      117      116      040
        013215      124      105      123
        013220      124      000
1793    013222      040      040      116  EM2202:: .ASCIZ /  NO TX_ACTION FOUND AFTER VALID DATA WORD TX'D ON LINE: /
        013225      117      040      124
        013230      130      137      101
        013233      103      124      111
        013236      117      116      040
        013241      106      117      125
        013244      116      104      040
        013247      101      106      124
        013252      105      122      040
        013255      126      101      114
        013260      111      104      040
        013263      104      101      124
        013266      101      040      127
        013271      117      122      104
        013274      040      124      130
        013277      047      104      040
        013302      117      116      040
        013305      114      111      116
        013310      105      072      040
        013313      000
1794    013314      040      040      111  EM2203:: .ASCIZ /  INCORRECT LINE NUMBER FOUND WITH TX_ACT AFTER DATA TX'D ON LINE : /
        013317      116      103      117
        013322      122      122      105
        013325      103      124      040
        013330      114      111      116
        013333      105      040      116
        013336      125      115      102
        013341      105      122      040
        013344      106      117      125
        013347      116      104      040
        013352      127      111      124
        013355      110      040      124
```

```
          013360  130  137  101
          013363  103  124  040
          013366  101  106  124
          013371  105  122  040
          013374  104  101  124
          013377  101  040  124
          013402  130  047  104
          013405  040  117  116
          013410  040  114  111
          013413  116  105  040
          013416  072  040  000
1795      013421  124  130  137  EM2301:: .ASCIZ /TX_ENABLE (INACTIVE) BIT TEST/
          013424  105  116  101
          013427  102  114  105
          013432  040  050  111
          013435  116  101  103
          013440  124  111  126
          013443  105  051  040
          013446  102  111  124
          013451  040  124  105
          013454  123  124  000
1796      013457  040  040  124  EM2302:: .ASCIZ /  TX_ENABLE BIT BAD ON LINE: /
          013462  130  137  105
          013465  116  101  102
          013470  114  105  040
          013473  102  111  124
          013476  040  102  101
          013501  104  040  117
          013504  116  040  114
          013507  111  116  105
          013512  072  040  000
1797      013515  124  130  137  EM2401:: .ASCIZ /TX_ENABLE (ACTIVE) BIT TEST/
          013520  105  116  101
          013523  102  114  105
          013526  040  050  101
          013531  103  124  111
          013534  126  105  051
          013537  040  102  111
          013542  124  040  124
          013545  105  123  124
          013550  000
1798      013551  122  105  103  EM2601:: .ASCIZ /RECEIVE INTERRUPT TEST /
          013554  105  111  126
          013557  105  040  111
          013562  116  124  105
          013565  122  122  125
          013570  120  124  040
          013573  124  105  123
          013576  124  040  000
1799      013601  040  040  116  EM2602:: .ASCIZ /  NO RX INT GENERATED (DATA_VALID SET, RX INTS ENABLED)./
          013604  117  040  122
          013607  130  040  111
          013612  116  124  040
          013615  107  105  116
          013620  105  122  101
          013623  124  105  104
          013626  040  050  104
```

```
        013631  101  124  101
        013634  137  126  101
        013637  114  111  104
        013642  040  123  105
        013645  124  054  040
        013650  122  130  040
        013653  111  116  124
        013656  123  040  105
        013661  116  101  102
        013664  114  105  104
        013667  051  056  000
1800    013672  040  040  116  EM2603:: .ASCIZ /  NO RX INT GENERATED (NO CODES IN FIFO AFTER RESET)./
        013675  117  040  122
        013700  130  040  111
        013703  116  124  040
        013706  107  105  116
        013711  105  122  101
        013714  124  105  104
        013717  040  050  116
        013722  117  040  103
        013725  117  104  105
        013730  123  040  111
        013733  116  040  106
        013736  111  106  117
        013741  040  101  106
        013744  124  105  122
        013747  040  122  105
        013752  123  105  124
        013755  051  056  000
1801    013760  040  040  116  EM2604:: .ASCIZ /  NO RX INT GENERATED (RX_DATA_AVAIL CLR, RX INTS ENABLED)./
        013763  117  040  122
        013766  130  040  111
        013771  116  124  040
        013774  107  105  116
        013777  105  122  101
        014002  124  105  104
        014005  040  050  122
        014010  130  137  104
        014013  101  124  101
        014016  137  101  126
        014021  101  111  114
        014024  040  103  114
        014027  122  054  040
        014032  122  130  040
        014035  111  116  124
        014040  123  040  105
        014043  116  101  102
        014046  114  105  104
        014051  051  056  000
1802    014054  040  040  122  EM2605:: .ASCIZ /  RX INTERRUPT GENERATED WITH RX_DATA_AVAIL CLEAR./
        014057  130  040  111
        014062  116  124  105
        014065  122  122  125
        014070  120  124  040
        014073  107  105  116
        014076  105  122  101
        014101  124  105  104
```

```
        014104  040  127  111
        014107  124  110  040
        014112  122  130  137
        014115  104  101  124
        014120  101  137  101
        014123  126  101  111
        014126  114  040  103
        014131  114  105  101
        014134  122  056  000
1803    014137  124  122  101  EM2606:: .ASCIZ /TRANSMIT INTERRUPT TEST ERROR:/
        014142  116  123  115
        014145  111  124  040
        014150  111  116  124
        014153  105  122  122
        014156  125  120  124
        014161  040  124  105
        014164  123  124  040
        014167  105  122  122
        014172  117  122  072
        014175  000
1804    014176  040  040  124  EM2607:: .ASCIZ /  TX_ACTION SET REPEATEDLY AFTER BOARD RESET, NO DATA SENT./
        014201  130  137  101
        014204  103  124  111
        014207  117  116  040
        014212  123  105  124
        014215  040  122  105
        014220  120  105  101
        014223  124  105  104
        014226  114  131  040
        014231  101  106  124
        014234  105  122  040
        014237  102  117  101
        014242  122  104  040
        014245  122  105  123
        014250  105  124  054
        014253  040  116  117
        014256  040  104  101
        014261  124  101  040
        014264  123  105  116
        014267  124  056  000
1805    014272  040  040  124  EM2608:: .ASCIZ /  TX_ACTION STUCK SET AFTER BOARD RESET./
        014275  130  137  101
        014300  103  124  111
        014303  117  116  040
        014306  123  124  125
        014311  103  113  040
        014314  123  105  124
        014317  040  101  106
        014322  124  105  122
        014325  040  102  117
        014330  101  122  104
        014333  040  122  105
        014336  123  105  124
        014341  056  000
1806    014343  040  040  124  EM2609:: .ASCIZ /  TX INTERRUPT GENERATED WITH TX_ACTION CLEAR./
        014346  130  040  111
        014351  116  124  105
```

```
        014354  122     122     125
        014357  120     124     040
        014362  107     105     116
        014365  105     122     101
        014370  124     105     104
        014373  040     127     111
        014376  124     110     040
        014401  124     130     137
        014404  101     103     124
        014407  111     117     116
        014412  040     103     114
        014415  105     101     122
        014420  056     000
1807    014422  040     040     116  EM2610:: .ASCIZ /  NO TX INTERRUPT WITH TX_ACTION SET AND TX INTS ENABLED./
        014425  117     040     124
        014430  130     040     111
        014433  116     124     105
        014436  122     122     125
        014441  120     124     040
        014444  127     111     124
        014447  110     040     124
        014452  130     137     101
        014455  103     124     111
        014460  117     116     040
        014463  123     105     124
        014466  040     101     116
        014471  104     040     124
        014474  130     040     111
        014477  116     124     123
        014502  040     105     116
        014505  101     102     114
        014510  105     104     056
        014513  000
1808    014514  040     040     124  EM2611:: .ASCIZ /  TX_ACTION NOT SET AFTER CHARS SENT ON ALL LINES./
        014517  130     137     101
        014522  103     124     111
        014525  117     116     040
        014530  116     117     124
        014533  040     123     105
        014536  124     040     101
        014541  106     124     105
        014544  122     040     103
        014547  110     101     122
        014552  123     040     123
        014555  105     116     124
        014560  040     117     116
        014563  040     101     114
        014566  114     040     114
        014571  111     116     105
        014574  123     056     000
1809    014577  111     116     124  EM3001:: .ASCIZ /INTERRUPT BR LEVEL TEST /
        014602  105     122     122
        014605  125     120     124
        014610  040     102     122
        014613  040     114     105
        014616  126     105     114
        014621  040     124     105
```

```
        014624  123  124  040
        014627  000
1810    014630  040  040  116  EM3002:: .ASCIZ /  NO RX_DATA_AVAIL FROM SELFTEST CODES IN FIFO AFTER RESET./
        014633  117  040  122
        014636  130  137  104
        014641  101  124  101
        014644  137  101  126
        014647  101  111  114
        014652  040  106  122
        014655  117  115  040
        014660  123  105  114
        014663  106  124  105
        014666  123  124  040
        014671  103  117  104
        014674  105  123  040
        014677  111  116  040
        014702  106  111  106
        014705  117  040  101
        014710  106  124  105
        014713  122  040  122
        014716  105  123  105
        014721  124  056  000
1811    014724  040  040  124  EM3003:: .ASCIZ /  TX INTERRUPT GENERATED AT WRONG BR LEVEL:/
        014727  130  040  111
        014732  116  124  105
        014735  122  122  125
        014740  120  124  040
        014743  107  105  116
        014746  105  122  101
        014751  124  105  104
        014754  040  101  124
        014757  040  127  122
        014762  117  116  107
        014765  040  102  122
        014770  040  114  105
        014773  126  105  114
        014776  072  000
1812    015000  040  040  122  EM3004:: .ASCIZ /  RX INTERRUPT GENERATED AT WRONG BR LEVEL:/
        015003  130  040  111
        015006  116  124  105
        015011  122  122  125
        015014  120  124  040
        015017  107  105  116
        015022  105  122  101
        015025  124  105  104
        015030  040  101  124
        015033  040  127  122
        015036  117  116  107
        015041  040  102  122
        015044  040  114  105
        015047  126  105  114
        015052  072  000
1813    015054  040  040  124  EM3005:: .ASCIZ /  TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS Rx INT./
        015057  130  040  111
        015062  116  124  105
        015065  122  122  125
        015070  120  124  040
```

```
        015073      107     111     126
        015076      105     116     040
        015101      120     122     105
        015104      103     105     104
        015107      105     116     103
        015112      105     040     117
        015115      126     105     122
        015120      040     123     111
        015123      115     125     114
        015126      124     101     116
        015131      105     117     125
        015134      123     040     122
        015137      130     040     111
        015142      116     124     056
        015145      000
1814    015146      104     111     101  EM3101:: .ASCIZ /DIAGNOSTIC FIELD (BMP) TEST/
        015151      107     116     117
        015154      123     124     111
        015157      103     040     106
        015162      111     105     114
        015165      104     040     050
        015170      102     115     120
        015173      051     040     124
        015176      105     123     124
        015201      000
1815    015202      040     040     104  EM3102:: .ASCIZ /  DIAGNOSTIC FIELD BAD ON LINE: /
        015205      111     101     107
        015210      116     117     123
        015213      124     111     103
        015216      040     106     111
        015221      105     114     104
        015224      040     102     101
        015227      104     040     117
        015232      116     040     114
        015235      111     116     105
        015240      072     040     000
1816    015243      105     130     120  EM9009:: .ASCIZ /EXPECTED OR CORRECT/
        015246      105     103     124
        015251      105     104     040
        01525.      117     122     040
        015257      103     117     122
        015262      122     105     103
        015265      124     000
1817    015267      101     103     124  EM9010:: .ASCIZ /ACTUAL OR MEASURED /
        015272      125     101     114
        015275      040     117     122
        015300      040     115     105
        015303      101     123     125
        015306      122     105     104
        015311      040     000
1818    015313      123     125     115  EM9014:: .ASCIZ /SUMMARY REPORTS FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:/
        015316      115     101     122
        015321      131     040     122
        015324      105     120     117
        015327      122     124     123
        015332      040     106     117
        015335      122     040     114
```

```
        015340      111     116     105
        015343      123     040     127
        015346      111     124     110
        015351      040     105     130
        015354      103     105     123
        015357      123     111     126
        015362      105     040     116
        015365      125     115     102
        015370      105     122     123
        015373      040     117     106
        015376      040     105     122
        015401      122     117     122
        015404      123     072     000
1819    015407      040     040     106  EM9017:: .ASCII /  FIFO WILL NOT PURGE (DATA.VALID STUCK SET),/
        015412      111     106     117
        015415      040     127     111
        015420      114     114     040
        015423      116     117     124
        015426      040     120     125
        015431      122     107     105
        015434      040     050     104
        015437      101     124     101
        015442      056     126     101
        015445      114     111     104
        015450      040     123     124
        015453      125     103     113
        015456      040     123     105
        015461      124     051     054
1820    015464      040     122     105          .ASCIZ / REMAINDER OF TEST SKIPPED./
        015467      115     101     111
        015472      116     104     105
        015475      122     040     117
        015500      106     040     124
        015503      105     123     124
        015506      040     123     113
        015511      111     120     120
        015514      105     104     056
        015517      000
1821    015520      116     117     040  EM9018:: .ASCIZ /NO CODE/
        015523      103     117     104
        015526      105     000
1822    015530      116     117     116  EM9019:: .ASCIZ /NON-SELFTEST/
        015533      055     123     105
        015536      114     106     124
        015541      105     123     124
        015544      000
1823    015545      123     105     114  EM9020:: .ASCIZ /SELFTEST ERROR CODE/
        015550      106     124     105
        015553      123     124     040
        015556      105     122     122
        015561      117     122     040
        015564      103     117     104
        015567      105     000
1824    015571      104     101     124  EM9022:: .ASCIZ /DATA CHARACTER/
        015574      101     040     103
        015577      110     101     122
        015602      101     103     124
```

```
                015605      105     122     000
1825            015610      115     117     104  EM9023:: .ASCIZ /MODEM STATUS CODE/
                015613      105     115     040
                015616      123     124     101
                015621      124     125     123
                015624      040     103     117
                015627      104     105     000
1826            015632      123     105     114  EM9024:: .ASCIZ /SELFTEST CODE/
                015635      106     124     105
                015640      123     124     040
                015643      103     117     104
                015646      105     000
1827            015650      040     040     040  EM9026:: .ASCIZ /     LPR CONTENTS: /
                015653      040     040     114
                015656      120     122     040
                015661      103     117     116
                015664      124     105     116
                015667      124     123     072
                015672      040     000
1828            015674      102     115     120  EM9301:: .ASCIZ /BMP CODE REPORT/
                015677      040     103     117
                015702      104     105     040
                015705      122     105     120
                015710      117     122     124
                015713      000
1829            015714      102     115     120  EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
                015717      040     103     117
                015722      104     105     040
                015725      106     117     125
                015730      116     104     040
                015733      111     116     040
                015736      124     105     123
                015741      124     040     000
1830            015744      124     110     105  EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
                015747      040     114     101
                015752      123     124     040
                015755      102     115     120
                015760      040     103     117
                015763      104     105     040
                015766      127     101     123
                015771      040     106     117
                015774      125     116     104
                015777      040     111     116
                016002      040     124     105
                016005      123     124     040
                016010      000
1831            016011      125     116     105  EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
                016014      130     120     105
                016017      103     124     105
                016022      104     040     102
                016025      115     120     040
                016030      103     117     104
                016033      105     123     040
                016036      106     117     125
                016041      116     104     040
                016044      104     125     122
                016047      111     116     107
```

GLOBAL MESSAGE AREA

```
        016052      040     124     110
        016055      111     123     040
        016060      120     101     123
        016063      123     000
  1832                                  .EVEN
```

```
1834
1835
1836                    ;*****************************************************************************************
1837                    ;
1838                    ;               FVTSKL2.P11
1839                    ;
1840                    ;*****************************************************************************************
1841
1842
1843
1844                    .SBTTL  GLOBAL ERROR REPORT SECTION
1845
1846                    ;++
1847                    ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
1848                    ; USED BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
1849                    ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
1850                    ;--
```

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

                            BGNMSG  ER0101
                                                                        ER0101::
                            SAVE                        ;SAVE THE GPR CONTENTS.
016066  004567  165704          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

016072  032705  000001      BIT     #BIT0,R5        ;TEST FOR READ ERROR.
016076  001410              BEQ     2$              ;SKIP READ ERROR MSG IF NO READ ERROR.
016100                      PRINTB  #MSG1           ;PRINT READ ERROR MESSAGE.
016100  012746  016172              MOV     #MSG1,-(SP)
016104  012746  000001              MOV     #1,-(SP)
016110  010600                      MOV     SP,R0
016112  104414                      TRAP    C$PNTB
016114  062706  000004              ADD     #4,SP
016120  032705  000002      2$:     BIT     #BIT1,R5        ;TEST FOR WRITE ERROR.
016124  001410              BEQ     4$              ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
016126                      PRINTB  #MSG2           ;PRINT WRITE ERROR MESSAGE.
016126  012746  016250              MOV     #MSG2,-(SP)
016132  012746  000001              MOV     #1,-(SP)
016136  010600                      MOV     SP,R0
016140  104414                      TRAP    C$PNTB
016142  062706  000004              ADD     #4,SP
016146          4$:     PRINTX  #MSG3           ;SUGGEST THAT DHV MAY BE AT WRONG ADDRESS.
016146  012746  016327              MOV     #MSG3,-(SP)
016152  012746  000001              MOV     #1,-(SP)
016156  010600                      MOV     SP,R0
016160  104415                      TRAP    C$PNTX
016162  062706  000004              ADD     #4,SP
                            PASS                        ;RESTORE THE GPR CONTENTS.
016166  004736                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                            ENDMSG
                                                                L10002:
016170  104423                                                      TRAP    C$MSG

016172      045     101     102 MSG1::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.%N/
```

```
        016175  125  123  040
        016200  124  111  115
        016203  105  055  117
        016206  125  124  040
        016211  124  122  101
        016214  120  040  103
        016217  101  125  123
        016222  105  104  040
        016225  102  131  040
        016230  122  105  101
        016233  104  040  101
        016236  124  124  105
        016241  115  120  124
        016244  056  045  116
        016247  000
1889    016250  045  101  102  MSG2::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.%N/
        016253  125  123  040
        016256  124  111  115
        016261  105  055  117
        016264  125  124  040
        016267  124  122  101
        016272  120  040  103
        016275  101  125  123
        016300  105  104  040
        016303  102  131  040
        016306  127  122  111
        016311  124  105  040
        016314  101  124  124
        016317  105  115  120
        016322  124  056  045
        016325  116  000
1890    016327  045  101  10%  MSG3::  .ASCIZ  /%ADHV MAY BE AT THE WRONG Q-BUS ADDRESS.%N%N/
        016332  110  126  040
        016335  115  101  131
        016340  040  102  105
        016343  040  101  124
        016346  040  124  110
        016351  105  040  127
        016354  122  117  116
        016357  107  040  121
        016362  055  102  125
        016365  123  040  101
        016370  104  104  122
        016373  105  123  123
        016376  056  045  116
        016401  045  116  000
1891
1892                                .EVEN
```

```
1894                            .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0201 -
1895                            ;*******************************************************************************
1896                            ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS 2 CONTIGUOUS
1897                            ;*      ASCII ERROR MESSAGES.  THE ADDRESS OF THE FIRST MESSAGE IS PASSED
1898                            ;*      AS AN INPUT PARAMETER AND THE ADDRESS OF THE SECOND IS FOUND BY
1899                            ;*      SEARCHING FOR THE END OF THE FIRST MESSAGE.
1900                            ;*
1901                            ;* INPUTS:      R1 - ADDRESS OF THE FIRST MESSAGE TO PRINT.
1902                            ;*
1903                            ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
1904                            ;*
1905                            ;* CALLING SEQUENCE:    LOAD THE ADDRESS OF THE FIRST MESSAGE IN R1.
1906                            ;*                      INCLUDE THE LABEL "ER0201" AS THE MESSAGE POINTER
1907                            ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1908                            ;*
1909                            ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
1910                            ;*              THE SECOND MESSAGE SHOULD FOLLOW THE FIRST ONE IN THE PROGRAM
1911                            ;*              MEMORY.  EACH MESSAGE SHOULD BE DEFINED USING .ASCIZ
1912                            ;*
1913                            ;* SUBORDINATE ROUTINES USED: NONE.
1914                            ;*******************************************************************************
1915
1916 016404                             BGNMSG  ER0201
     016404                                                                 ER0201::
1917 016404                             SAVE                    ;SAVE THE GPR CONTENTS.
     016404  004567  165366                     JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
1918
1919 016410  010102                      MOV     R1,R2
1920 016412  105722                2$:   TSTB    (R2)+           ;CHECK FOR A ZERO BYTE (END OF MESSAGE).
1921 016414  001376                      BNE     2$              ;LOOP UNTIL NEXT MESSAGE IS FOUND.
1922
1923 016416                             PRINTB  #EF0503,R1      ;PRINT THE FIRST MESSAGE.
     016416  010146                                                     MOV     R1,-(SP)
     016420  012746  004132                                             MOV     #EF0503,-(SP)
     016424  012746  000002                                             MOV     #2,-(SP)
     016430  010600                                                     MOV     SP,R0
     016432  104414                                                     TRAP    C$PNTB
     016434  062706  000006                                             ADD     #6,SP
1924 016440                             PRINTB  #EF0503,R2      ;PRINT THE SECOND MESSAGE.
     016440  010246                                                     MOV     R2,-(SP)
     016442  012746  004132                                             MOV     #EF0503,-(SP)
     016446  012746  000002                                             MOV     #2,-(SP)
     016452  010600                                                     MOV     SP,R0
     016454  104414                                                     TRAP    C$PNTB
     016456  062706  000006                                             ADD     #6,SP
1925
1926 016462                             PASS                    ;RESTORE THE GPR CONTENTS.
     016462  004736                             JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
1927
1928 016464                             ENDMSG
     016464                                                             L10003:
     016464  104423                                                             TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0503 -
                        ;*****************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
                        ;*      MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF THE MESSAGE TO PRINT.
                        ;*
                        ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    LOAD THE ADDRESS OF THE MESSAGE IN R1.
                        ;*                      INCLUDE THE LABEL "ER0503" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*****************************************************************************

016466                          BGNMSG  ER0503
016466                                                                  ER0503::

                                PRINTB  #EF0503,R1      ;PRINT THE MESSAGE.
016466  010146                                                          MOV     R1,-(SP)
016470  012746  004132                                                  MOV     #EF0503,-(SP)
016474  012746  000002                                                  MOV     #2,-(SP)
016500  010600                                                          MOV     SP,R0
016502  104414                                                          TRAP    C$PNTB
016504  062706  000006                                                  ADD     #6,SP

016510                          ENDMSG
016510                                                                  L10004:
016510  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0504 -
                        ;*****************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
                        ;*      MESSAGES WHEN ILLEGAL INTERRUPTS ARE RECEIVED.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF THE MESSAGE TO PRINT.
                        ;*              R2 - NUMBER OF ILLEGAL INTEPRUPTS RECEIVED.
                        ;*
                        ;* OUTPUTS:     MESSAGESS ARE PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    LOAD THE ADDRESS OF THE MESSAGE IN R1.
                        ;*                      LOAD THE NUMBER OF ILLEGAL INTS IN R2.
                        ;*                      INCLUDE THE LABEL "ER0504" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*****************************************************************************

                              BGNMSG  ER0504
016512                                                          ER0504::
                        
016512                        PRINTB  #EF0503,R1      ;PRINT THE FIRST LINE OF THE MESSAGE.
016512  010146                                                  MOV     R1,-(SP)
016514  012746  004132                                          MOV     #EF0503,-(SP)
016520  012746  000002                                          MOV     #2,-(SP)
016524  010600                                                  MOV     SP,R0
016526  104414                                                  TRAP    C$PNTB
016530  062706  000006                                          ADD     #6,SP
016534                        PRINTX  #EF0505,R2      ;PRINT THE NUMBER OF INTS RECEIVED.
016534  010246                                                  MOV     R2,-(SP)
016536  012746  004137                                          MOV     #EF0505,-(SP)
016542  012746  000002                                          MOV     #2,-(SP)
016546  010600                                                  MOV     SP,R0
016550  104415                                                  TRAP    C$PNTX
016552  062706  000006                                          ADD     #6,SP

                              ENDMSG
016556                                                  L10005:
016556  104423                                                  TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER1401 -
                        ;*************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
                        ;*      INFORMATION IF AN ERROR IS DETECTED IN THE ROM VERSION TEST.
                        ;*      THIS SUBROUTINE ANALYSES THE INPUT PARAMETERS WHICH CONTAIN THE
                        ;*      ROM VERSION NUMBERS FOR PROC_1 AND PROC_2 AND REPORTS THE APPROPRIATE
                        ;*      ERROR MESSAGE TO THE OPERATOR.
                        ;*
                        ;* INPUTS:      R1 - CONTAINS THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
                        ;*              R3 - CONTAINS THE ROM VERSION NUMBER OF PROC_1.
                        ;*              R4 _ CONTAINS THE ROM VERSION NUMBER OF PROC_2.
                        ;*
                        ;* OUTPUTS:     BASIC AND EXTENDED ERROR MESSAGES ARE REPORTED, AT THE
                        ;*              OPERATORS CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER1401" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*************************************************************************

                            BGNMSG  ER1401
016560                                                          ER1401::
                        
                            PRINTB  #EF0503,R1          ;REPORT THE ERROR MESSAGE PASSED IN.
016560  010146                                                  MOV     R1,-(SP)
016562  012746  004132                                          MOV     #EF0503,-(SP)
016566  012746  000002                                          MOV     #2,-(SP)
016572  010600                                                  MOV     SP,R0
016574  104414                                                  TRAP    C$PNTB
016576  062706  000006                                          ADD     #6,SP
                        
                        ;+
                        ; DETERMINE WHICH ROM VERSION NUMBER(S) ARE MISSING.
                        ;-
                        
016602  012705  000143          MOV     #99.,R5             ;GET INVALID ROM NUMBER.
016606  012701  012312          MOV     #EM1405,R1          ;SELECT PROC_1 MESSAGE.
016612  012702  012340          MOV     #EM1407,R2          ;SELECT THE "NOT FOUND" MESSAGE.
016616  120305                  CMPB    R3,R5               ;CHECK PROC_1 ROM VERSION NUMBER.
016620  001402                  BEQ     2$                  ;GO REPORT PROC_1 CODE NOT FOUND.
016622  012702  012352          MOV     #EM1408,R2          ;SELECT "FOUND" MESSAGE.
016626  004767  000026  2$:     JSR     PC,50$              ;GO REPORT MESSAGE.
                        
016632  012701  012325          MOV     #EM1406,R1          ;SELECT PROC_2 MESSAGE.
016636  012702  012340          MOV     #EM1407,R2          ;SELECT THE "NOT FOUND" MESSAGE.
016642  120405                  CMPB    R4,R5               ;CHECK PROC_2 ROM VERSION NUMBER.
016644  001402                  BEQ     4$                  ;GO REPORT PROC_2 CODE NOT FOUND.
016646  012702  012352          MOV     #EM1408,R2          ;SELECT "FOUND" MESSAGE.
016652  004767  000002  4$:     JSR     PC,50$              ;GO REPORT THE MESSAGE.
016656  000413                  BR      60$                 ;EXIT.
                        
016660                  50$:    PRINTX  #EF1402,R1,R2       ;REPORT THE MESSAGE.
016660  010246                                                  MOV     R2,-(SP)
016662  010146                                                  MOV     R1,-(SP)
```

```
        016664  012746  004314                                          MOV     #EF1402,-(SP)
        016670  012746  000003                                          MOV     #3,-(SP)
        016674  010600                                                  MOV     SP,R0
        016676  104415                                                  TRAP    C$PNTX
        016700  062706  000010                                          ADD     #10,SP
  2029  016704  000207                  RTS     PC              ;RETURN.
  2030  016706                  60$:    ENDMSG
        016706                                                  L10006:
        016706  104423                                                  TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER1601 -
                        ;*****************************************************************************
                        ;*      THIS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
                        ;*      INFORMATION IF AN ERROR IS DETECTED IN ONE OF THE DEVICE REGISTER
                        ;*      ACCESS TESTS.
                        ;*      THIS SUBROUTINE REPORTS THE ACTUAL AND EXPECTED FROM THE DEVICE
                        ;*      REGISTER(S) WHICH IS(ARE) IN FAULTY
                        ;*
                        ;* INPUTS:      R1 - ACTUAL DATA (UNUSED BITS SET TO 0).
                        ;*              R2 - EXPECTED DATA (UNUSED BITS SET TO 0).
                        ;*              R3 - OFFSET (IN BYTES) TO THE REGISTER BEING TESTED.
                        ;*              R5 - LINE NUMBER OF REGISTER BEING TESTED.
                        ;*              RMATBB - LABEL AT BASE OF REGISTER MESSAGE ADDRESS TABLE.
                        ;*
                        ;* OUTPUTS:     MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABLE "ER1601" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE
                        ;*****************************************************************************
                              BGNMSG  ER1601
016710                                                          ER1601::
016710
016710  016304  002304          MOV     RMATBB(R3),R4   ;FETCH ADDRESS OF REGISTER NAME MESSAGE.
016714                        PRINTB  #EF1604,R4,R5   ;REPORT BASIC MESSAGE (REG NAME AND LINE #).
016714  010546                                                  MOV     R5,-(SP)
016716  010446                                                  MOV     R4,-(SP)
016720  012746  004501                                          MOV     #EF1604,-(SP)
016724  012746  000003                                          MOV     #3,-(SP)
016730  010600                                                  MOV     SP,R0
016732  104414                                                  TRAP    C$PNTB
016734  062706  000010                                          ADD     #10,SP
016740                        PRINTX  #EF1602,R2      ;PRINT THE EXPECTED DATA.
016740  010246                                                  MOV     R2,-(SP)
016742  012746  004375                                          MOV     #EF1602,-(SP)
016746  012746  000002                                          MOV     #2,-(SP)
016752  010600                                                  MOV     SP,R0
016754  104415                                                  TRAP    C$PNTX
016756  062706  000006                                          ADD     #6,SP
016762                        PRINTX  #EF1603,R1      ;PRINT THE ACTUAL DATA.
016762  010146                                                  MOV     R1,-(SP)
016764  012746  004437                                          MOV     #EF1603,-(SP)
016770  012746  000002                                          MOV     #2,-(SP)
016774  010600                                                  MOV     SP,R0
016776  104415                                                  TRAP    C$PNTX
017000  062706  000006                                          ADD     #6,SP
017004                        ENDMSG
017004                                                  L10007:
017004  104423                                                  TRAP    C$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.
                        ;*************************************************************************
017006                          BGNMSG  ER1603
017006                                                                  ER1603::
017006                          SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
017006  004567  164764                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

017012                           PRINTB #EF0503,R1      ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
017012  010146                                                          MOV     R1,-(SP)
017014  012746  004132                                                  MOV     #EF0503,-(SP)
017020  012746  000002                                                  MOV     #2,-(SP)
017024  010600                                                          MOV     SP,R0
017026  104414                                                          TRAP    C$PNTB
017030  062706  000006                                                  ADD     #6,SP

017034  016702  164732          MOV     ERRMSG,R2       ;GET THE "TEST MESSAGE".
017040                          PRINTB #EF1601,R2       ;PRINT "TEST ABORTED" MESSAGE.
017040  010246                                                          MOV     R2,-(SP)
017042  012746  004351                                                  MOV     #EF1601,-(SP)
017046  012746  000002                                                  MOV     #2,-(SP)
017052  010600                                                          MOV     SP,R0
017054  104414                                                          TRAP    C$PNTB
017056  062706  000006                                                  ADD     #6,SP

017062                          PASS                    ;RESTORE THE CONTENTS OF THE GPRS.
017062  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.

017064                          ENDMSG
017064                                                                  L10010:
017064  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER3001 -
                        ;*************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS INTENDED FOR USE IN THE
                        ;*      INTERRUPT BR LEVEL TEST.  IT REPORTS ADDITIONAL INFORMATION WHEN AN
                        ;*      INTERRUPT HAS OCCURRED AT THE WRONG BR LEVEL.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
                        ;*              R4 - BR LEVEL AT WHICH THE INT REQUEST OCCURRED.
                        ;*              R5 - EXPECTED OR CORRECT BR LEVEL FOR THE DUT.
                        ;*
                        ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER3001" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*************************************************************************

017066                          BGNMSG  ER3001
017066                                                                  ER3001::
017066                          PRINTB  #EF0503,R1      ;PRINT THE FIRST LINE OF THE MESSAGE.
017066  010146                                                          MOV     R1,-(SP)
017070  012746  004132                                                  MOV     #EF0503,-(SP)
017074  012746  000002                                                  MOV     #2,-(SP)
017100  010600                                                          MOV     SP,R0
017102  104414                                                          TRAP    C$PNTB
017104  062706  000006                                                  ADD     #6,SP
017110                          PRINTX  #EF3001,R5      ;REPORT EXPECTED BR LEVEL.
017110  010546                                                          MOV     R5,-(SP)
017112  012746  004576                                                  MOV     #EF3001,-(SP)
017116  012746  000002                                                  MOV     #2,-(SP)
017122  010600                                                          MOV     SP,R0
017124  104415                                                          TRAP    C$PNTX
017126  062706  000006                                                  ADD     #6,SP
017132                  PRINTX  #EF3002,R4      ;REPORT ACTUAL BR LEVEL.
017132  010446                                                          MOV     R4,-(SP)
017134  012746  004645                                                  MOV     #EF3002,-(SP)
017140  012746  000002                                                  MOV     #2,-(SP)
017144  010600                                                          MOV     SP,R0
017146  104415                                                          TRAP    C$PNTX
017150  062706  000006                                                  ADD     #6,SP

017154                  ENDMSG
017154                                                          L10011:
017154  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9004 -
                        ;*****************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH REPORTS ERROR SUMMARIES
                        ;*      FOR LINES WHICH HAVE EXCEEDED THE SPECIFIED MAXIMUM NUMBER OF
                        ;*      INDIVIDUAL RECEPTION ERRORS.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
                        ;*              ERCNTB - LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
                        ;*              ERSMRF - "REPORT ERROR SUMMARY FOR LINE" FLAGS.
                        ;*
                        ;* OUTPUTS:     A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9004" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
                        ;*              THE CONTENTS OF GPR'S R2, R3, R4, AND R5 ARE DESTROYED.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*****************************************************************************

017156                          BGNMSG  ER9004
017156                                                                  ER9004::

                        PRINTB  #EF0503,#EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
017156  012746  015313                                                  MOV     #EM9014,-(SP)
017162  012746  004132                                                  MOV     #EF0503,-(SP)
017166  012746  000002                                                  MOV     #2,-(SP)
017172  010600                                                          MOV     SP,R0
017174  104414                                                          TRAP    C$PNTB
017176  062706  000006                                                  ADD     #6,SP
017202  005002                  CLR     R2              ;CLEAR THE LINE COUNTER.
017204  016703  163252          MOV     ERSMRF,R3       ;GET THE ERROR SUMMARY FLAGS.
017210  005004                  CLR     R4              ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
017212  000241          2$:     CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
017214  006003                  ROR     R3              ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
017216  103013                  BCC     4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
017220                          PRINTX  #EF9010,R2,ERCNTB(R4)
017220  016446  002464                                                  MOV     ERCNTB(R4),-(SP)
017224  010246                                                          MOV     R2,-(SP)
017226  012746  005223                                                  MOV     #EF9010,-(SP)
017232  012746  000003                                                  MOV     #3,-(SP)
017236  010600                                                          MOV     SP,R0
017240  104415                                                          TRAP    C$PNTX
017242  062706  000010                                                  ADD     #10,SP
017246  012405          4$:     MOV     (R4)+,R5        ;INCREMENT THE LINE OFFSET BY 2.
017250  005202                  INC     R2              ;INCREMT THE LINE COUNTER.
017252  005703                  TST     R3              ;CHECK THE ERROR SUMMARY FLAGS.
017254  001356                  BNE     2$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.

017256                          ENDMSG
017256                                                                  L10012:
017256  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9007 -
                        ;*************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS USED TO REPORT THAT
                        ;*      SOMETHING OTHER THAN A SELFTEST CODE WAS FOUND IN A SELFTEST CODE
                        ;*      FIFO SLOT DURING THE REMOVAL OF THE SELFTEST CODES FROM THE FIFO.
                        ;*      THIS ROUTINE IS USED BY THE RSTRPT ROUTINE.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF ERROR MESSAGE QUALIFIER STRING.
                        ;*              R2 - INCORRECT CODE AS READ FROM THE SELFTEST CODE FIFO SLOT.
                        ;*              R3 - LINE NUMBER ASSOCIATED WITH THE SELFTEST FIFO SLOT.
                        ;*
                        ;* OUTPUTS:     A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9007" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*************************************************************************

 017260                         BGNMSG  ER9007
 017260                                                                 ER9007::
                        
 017260  042703  177760          BIC     #177760,R3      ;REMOVE ALL BUT LINE # BITS FROM LINE # WORD.
 017264                         PRINTB  #EF9018,R1,R3   ;REPORT SECONDARY ERROR MESSAGE.
 017264  010346                                                         MOV     R3,-(SP)
 017266  010146                                                         MOV     R1,-(SP)
 017270  012746  005473                                                 MOV     #EF9018,-(SP)
 017274  012746  000003                                                 MOV     #3,-(SP)
 017300  010600                                                         MOV     SP,R0
 017302  104414                                                         TRAP    C$PNTB
 017304  062706  000010                                                 ADD     #10,SP
 017310                         PRINTX  #EF9017,R1,R2   ;REPORT THE ACTUAL INCORRECT CODE.
 017310  010246                                                         MOV     R2,-(SP)
 017312  010146                                                         MOV     R1,-(SP)
 017314  012746  005417                                                 MOV     #EF9017,-(SP)
 017320  012746  000003                                                 MOV     #3,-(SP)
 017324  010600                                                         MOV     SP,R0
 017326  104415                                                         TRAP    C$PNTX
 017330  062706  000010                                                 ADD     #10,SP
                        
 017334                         ENDMSG
 017334                                                         L10013:
 017334  104423                                                         TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9008 -
                        ;*************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS USED TO REPORT THAT
                        ;*      AN UNEXPECTED CODE OR CHARACTER HAS BEEN FOUND IN THE DUT RECEIVE
                        ;*      CHARACTER FIFO.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF PARTIAL ERROR MESSAGE STRING.
                        ;*              R2 - INCORRECT CODE AS READ FROM THE SELFTEST CODE FIFO SLOT.
                        ;*
                        ;* OUTPUTS:     A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9008" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*************************************************************************

                              BGNMSG  ER9008
017336                                                                  ER9008::

                        ;*
                        ; EXTRACT THE LINE NUMBER FROM THE INCORRECT CODE OR CHARACTER WHICH WAS READ
                        ;    FROM THE SELFTEST CODE FIFO SLOT.
                        ;-
017336  010203                 MOV     R2,R3
017340  000303                 SWAB    R3
017342  042703  177760         BIC     #177760,R3      ;CALCULATE LINE NUMBER OF CODE.
017346                  PRINTB  #EF9016,R1,R3   ;REPORT TYPE OF INCORRECT CODE FOUND.
017346  010346                                                          MOV     R3,-(SP)
017350  010146                                                          MOV     R1,-(SP)
017352  012746  005322                                                 MOV     #EF9016,-(SP)
017356  012746  000003                                                 MOV     #3,-(SP)
017362  010600                                                          MOV     SP,R0
017364  104414                                                          TRAP    C$PNTB
017366  062706  000010                                                 ADD     #10,SP
017372                  PRINTX  #EF9017,R1,R2   ;REPORT THE ACTUAL INCORRECT CODE.
017372  010246                                                          MOV     R2,-(SP)
017374  010146                                                          MOV     R1,-(SP)
017376  012746  005417                                                 MOV     #EF9017,-(SP)
017402  012746  000003                                                 MOV     #3,-(SP)
017406  010600                                                          MOV     SP,R0
017410  104415                                                          TRAP    C$PNTX
017412  062706  000010                                                 ADD     #10,SP

017416                  ENDMSG
017416                                                          L10014:
017416  104423                                                          TRAP    C$MSG
```

```
                        .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
017420                                                                  ER9101::

                              PRINTB  #EF9006,R2,R1   ;REPORT THE STRING FOLLOWED BY THE NUMBER.
017420  010146                                                          MOV     R1,-(SP)
017422  010246                                                          MOV     R2,-(SP)
017424  012746  005177                                                  MOV     #EF9006,-(SP)
017430  012746  000003                                                  MOV     #3,-(SP)
017434  010600                                                          MOV     SP,R0
017436  104414                                                          TRAP    C$PNTB
017440  062706  000010                                                  ADD     #10,SP

017444                  ENDMSG                                  L10015:
017444                                                                  TRAP    C$MSG
017444  104423
```

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

                            BGNMSG  ER9301
017446                                                          ER9301::
                            SAVE                        ;SAVE THE GPRS ON THE STACK.
017446  004567  164324              JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

                            PRINTB  #EF0503,R1          ;REPORT UNEXPECTED BMP CODES FOUND.
017452  010146                                                  MOV     R1,-(SP)
017454  012746  004132                                          MOV     #EF0503,-(SP)
017460  012746  000002                                          MOV     #2,-(SP)
017464  010600                                                  MOV     SP,R0
017466  104414                                                  TRAP    C$PNTB
017470  062706  000006                                          ADD     #6,SP
                            MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
                            MOV     #EM9302,R5      ;GET THE MESSAGE TO BE REPORTED.
                    2$:     MOV     (R3)+,R1        ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
                            MOV     (R3)+,R4        ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
                            JSR     PC,50$          ;GO REPORT THE BMP CODE.
                            CMP     R3,R2           ;CHECK IF ALL CODES HAVE BEEN REPORTED.
                            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.
                        ;-
                            CMP     R2,#BMPCQE-4    ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
                            BNE     60$             ;EXIT IF NOT AT THE LAST LOCATION.
                            TST     2(R2)           ;CHECK FOR A BMP CODE IN THE LAST CELL
                            BEQ     60$             ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
                            MOV     (R3)+,R1        ;GET THE TEST NUMBER OFF THE QUEUE.
                            MOV     (R3),R4         ;GET THE BMP CODE OFF THE QUEUE.
                            MOV     #EM9303,R5      ;SELECT THE MESSAGE TO BE REPORTED.
                            PRINTX  #EF9302         ;REPORT OVERFLOW CONDITION.
017544  012746  005702                                          MOV     #EF9302,-(SP)
017550  012746  000001                                          MOV     #1,-(SP)
017554  010600                                                  MOV     SP,R0
017556  104415                                                  TRAP    C$PNTX
```

```
     017560  062706  000004                                                    ADD     #4,SP
2294 017564  004767  000002            JSR     PC,50$          ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
2295 017570  000414                    BR      60$             ;EXIT.
2296
2297 017572                    50$:    PRINTX  #EF9301,R5,R1,R4  ;PRINT THE MESSAGE.
     017572  010446                                                            MOV     R4,-(SP)
     017574  010146                                                            MOV     R1,-(SP)
     017576  010546                                                            MOV     R5,-(SP)
     017600  012746  005624                                                    MOV     #EF9301,-(SP)
     017604  012746  000004                                                    MOV     #4,-(SP)
     017610  010600                                                            MOV     SP,R0
     017612  104415                                                            TRAP    C$PNTX
     017614  062706  000012                                                    ADD     #12,SP
2298 017620  000207                    RTS     PC              ;RETURN.
2299 017622                    60$:    PASS                    ;RESTORE THE GPR CONTENTS.
     017622  004736                            JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
2300
2301 017624                    ENDMSG
     017624                                                            L10016:
     017624  104423                                                            TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL SUBROUTINES SECTION
                        ;*****************************************************************************
                        ;
                        ;               FVTSKL3.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.
                        ;-- ************************************************************************

017626                  ALTFLD:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017626  004567  164144                 JSR     R5,PREGOS       ;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.
                        ;-
017632  010400                      MOV     R4,R0           ;CALCULATE THE NEW CONTENTS OF THE
017634  005100                      COM     R0              ;  REGISTER FIELDS WHICH ARE TO BE
017636  040002                      BIC     R0,R2           ;  ALTERRED BY THIS ROUTINE.
017640  016705  162464              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).
                        ;-
017644  000241                      CLC                     ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
017646  006003            2$:       ROR     R3              ;GET THE LINE SELECT BIT FOR THIS LINE.
017650  103006                    BCC     4$                ;SKIP SETUP IF LINE IS NOT SELECTED.
017652  010577  162366              MOV     R5,@CSRA        ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
017656  011100                      MOV     (R1),R0         ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
017660  040400                      BIC     R4,R0           ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
017662  050200                      BIS     R2,R0           ;OR IN THE NEW STATES OF THE FIELDS.
017664  010011                      MOV     R0,(R1)         ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
017666  005205            4$:       INC     R5              ;SET LINE NUMBER TO THE NEXT LINE.
017670  005703                      TST     R3              ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
017672  001365                    BNE     2$                ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.
```

```
2374
2375 017674                    60$:    PASS                    ;RESTORE GPRS.
     017674  004736                            JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
2376 017676  000207                    RTS     PC              ;RETURN TO CALLING ROUTNE.
```

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

017700                          CALMSL:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
017700  004567  164072                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
017704  005067  000210                  CLR     62$             ;CLEAR THE 2ND TIME FLAG.
                                ;*
                                ; SYNCHRONIZE WITH THE LTC.
                                ;-
017710  012705  000001          2$:     MOV     #1,R5           ;SET OUTER LOOP COUNTER TO 1 LOOP.
                                                                ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE <**
                                                                ;FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <**
017714  005000                          CLR     R0              ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
017716  012767  000001  162440          MOV     #1,TIMER1       ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
017724  005767  162434          4$:     TST     TIMER1          ;CHECK FOR COUNTER HAVING GONE TO ZERO.
017730  001410                          BEQ     6$              ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
017732  005200                          INC     R0              ;COUNT THIS ITERATION OF THE INNER LOOP.
017734  001373                          BNE     4$              ;LOOP IF COUNTER HAS NOT TURNED OVER.
017736  005305                          DEC     R5              ;DECREMENT THE INNER LOOP COUNTER.
017740  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.
                                ;-
017742  005067  162414                  CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
017746  000241                          CLC                     ;INDICATE FAILURE FOR RETURN.
017750  000461                          BR      60$             ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
                                ;*
                                ; WE ARE NOW SYNCHRONIZED WITH THE LTC.
                                ; SET UP FOR THE CALIBRATION LOOP.
                                ;-
017752  012704  002364          6$:     MOV     #TIMER1,R4      ;WILL TEST TIMER1 IN THE LOOP BELOW.
```

```
2434 017756  005001                        CLR     R1              ;CLEAR THE OUTER LOOP COUNTER.
2435 017760  005002                        CLR     R2              ;INDICATE TO CHECK ALL BITS OF TIMER1.
2436 017762  005003                        CLR     R3              ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2437 017764  012714  000001                MOV     #1,(R4)         ;LOAD TIMER1 WITH COUNT OF 1.
2438
2439 017770  016705  162402        8$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2440 017774  011400                10$:    MOV     (R4),R0         ;GET THE TIMER1 VALUE.
2441 017776  010067  000120                MOV     R0,64$          ;SAVE WORD (LIKE IN THE REAL LOOP).
2442 020002  040200                        BIC     R2,R0           ;LEAVE ALL THE BITS.
2443 020004  020003                        CMP     R0,R3           ;COMPARE AGAINST ZERO.
2444 020006  000261                        SEC                     ;SET CARRY IN CASE OF SUCCESS.
2445 020010  001406                        BEQ     12$             ;EXIT LOOP IF TIMER1 HAS CLEARED.
2446 020012  005305                        DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2447 020014  001367                        BNE     10$             ;LOOP IF MS NOT UP.
2448 020016  005301                        DEC     R1              ;DECREMENT THE MS TIME COUNT.
2449 020020  001363                        BNE     8$              ;KEEP LOOPING.
2450 020022  004767  000440                JSR     PC,OOPS         ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
2451                                ;+
2452                                ; WE HAVE NOW HAVE LOOP COUNT INFORMATION FOR ONE CLOCK TICK.
2453                                ; WE HAVE NEGATIVE OF NUMBER OF OUTER LOOPS IN R1, EACH IS MSLCNT INNER LOOPS.
2454                                ; WE HAVE THE PORTION OF THE LAST OUTER LOOP NOT EXECUTED, IN R5.
2455                                ; NOW WE CALCULATE THE TOTAL NUMBER OF INNER LOOPS EXECUTED.
2456                                ;-
2457 020026  005401                12$:    NEG     R1              ;GET NUMBER OF OUTER LOOPS.
2458 020030  016702  162342                MOV     MSLCNT,R2       ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2459 020034  010203                        MOV     R2,R3           ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2460 020036  160502                        SUB     R5,R2           ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2461 020040  010204                        MOV     R2,R4           ;  AND ADD TO ACCUMULATOR LSWORD.
2462 020042  005005                        CLR     R5              ;CLEAR ACCUMULATOR MSWORD.
2463 020044  005301                14$:    DEC     R1              ;CHECK R1 FOR 0 CONDITION
2464 020046  100403                        BMI     16$             ; SKIP MULTIPLICATION IF ZERO
2465 020050  060304                        ADD     R3,R4           ;MULTIPLY NUMBER OF INNER
2466 020052  005505                        ADC     R5              ; LOOPS PER OUTER LOOP BY
2467 020054  000773                        BR      14$             ;NUMBER OF OUTER LOOPS PERFORMED.
2468                                ;+
2469                                ; DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER LTC TICK.
2470                                ;-
2471 020056  016701  162312        16$:    MOV     MSTICK,R1       ;# OF MS PER LTC TICK IS DIVISOR.
2472 020062  010403                        MOV     R4,R3           ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2473 020064  010502                        MOV     R5,R2           ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2474 020066  004767  002756                JSR     PC,UNSDIV       ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
2475 020072  103402                        BCS     18$             ;BYPASS OOPS IF WE'RE OK.
2476 020074  004767  000366                JSR     PC,OOPS         ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2477 020100  010167  162272        18$:    MOV     R1,MSLCNT       ;SET NEW VALUE FOR MS LOOP COUNT.
2478 020104  005167  000010                COM     62$             ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2479 020110  001277                        BNE     2$              ;BRANCH IF ONLY ONE ITERATION DONE.
2480 020112  000261                        SEC                     ;SET THE SUCCESS FLAG FOR EXIT.
2481
2482 020114                        60$:    PASS                    ;RESTORE GPRS.
     020114  004736                                JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
2483 020116  000207                        RTS     PC              ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
2484
2485 020120  000000                62$:    .WORD   0               ;2ND CALIBRATION ITERATION FLAGS.
2486 020122  000000                64$:    .WORD   0               ;DUMMY WORD FOR STORAGE OF THE READ WORD.
```

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

020124                  CKTRAP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020124  004567  163646                      JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
020130  005067  162212          CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAGS.
020134  011011                  MOV     (R0),(R1)       ;PERFORM THE MOVE IN QUESTION.
020136  005767  162204  ADRPTR:: TST     TP4FLG          ;CHECK FOR OCCURENCE OF TRAP.
020142  000261                  SEC                     ;INDICATE SUCCESS.
020144  001401                  BEQ     60$             ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
020146  000241                  CLC                     ;INDICATE FAILURE.
020150                  60$:    PASS                    ;RESTORE GPRS.
020150  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
020152  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                       - CLNRST -
                        ;*******************************************************************************
                        ;*              - CLEAN RESET OF THE DEVICE UNDER TEST -
                        ;*      THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
                        ;*      THE DUT'S SELF-TEST IS SKIPPED,AND THE FIFO IS PURGED OF ANY ERROR
                        ;*      CODES, ETC.
                        ;*      IF THE RESET DOES NOT SUCCESFULLY COMPLETE, THEN THE CARRY BIT IS
                        ;*      PASSED BACK TO THE CALLING ROUTINE (CLEAR).
                        ;*
                        ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR
                        ;*              TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
                        ;*              ERRNBR - ERROR NUMBER FOR POSSIBLE ERROR REPORT.
                        ;*              ERRTBL- ERRTYP,ERNBR,AND ERRMSG SET UP CORRECTLY.
                        ;*
                        ;* OUTPUTS:     THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
                        ;*              CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
                        ;*              ERRBLK - VALUE MAY BE DESTROYED.
                        ;*              IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
                        ;*              TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CLNRST
                        ;*
                        ;* COMMENTS:    THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS ERRNBR.
                        ;*              THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET,PUFIFO,RESETT.
                        ;*******************************************************************************

020154                  CLNRST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020154  004567  163616                 JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; RESET THE DUT.
                        ;  THIS ROUTINE REPORTS ERRORS WITH NUMBERS FROM ERRNBR THRU ERRNBR+2.
                        ;-
020160  004767  001270          JSR     PC,RESETT       ;RESET THE DUT TO A KNOWN STATE.
020164  103002                  BCC     60$             ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
                        ;*
                        ; PURGE THE FIFO OF ERROR CODES, SAVE ANY BMP CODES FOUND.
                        ;-
020166  004767  000522          JSR     PC,PUFIFO       ;PURGE THE FIFO.

020172                  60$:                            ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
020172                          PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
020172  004736                          JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
                                                        ;CARRY BIT:IF CLEAR, THEN ABORT THE TEST.
020174  000207                  RTS     PC
```

```
2566                                    .SBTTL  GLOBAL SUBROUTINE                       - CLR16W -
2567                                    ;** ************************************************************************
2568                                    ;*                          - CLEAR SIXTEEN WORDS ROUTINE -
2569                                    ;*      THIS SUBROUTINE CLEARS 16 WORDS STARTING WITH THE SPECIFIED WORD.
2570                                    ;*
2571                                    ;* INPUTS:      R0 - ADDRESS OF THE FIRST WORD TO CLEAR.
2572                                    ;*
2573                                    ;* OUTPUTS:     (R0) TO (R0+15) - 16 WORDS OF MEMORY ARE CLEARED TO 0.
2574                                    ;*
2575                                    ;* CALLING SEQUENCE:    JSR     PC,CLR16W
2576                                    ;*
2577                                    ;* COMMENTS:
2578                                    ;*
2579                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
2580                                    ;-- ************************************************************************
2581
2582 020176                            CLR16W:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     020176  004567  163574                      JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
2583 020202  012701  000020                      MOV     #16.,R1         ;SET THE LOOP COUNTER TO 16.
2584 020206  005020                    2$:       CLR     (R0)+           ;CLEAR A WORD OF MEMORY.
2585 020210  005301                              DEC     R1              ;COUNT THIS LOOP.
2586 020212  001375                              BNE     2$              ;LOOP IF NOT 16 WORD CLEARED.
2587 020214                            60$:      PASS                    ;RESTORE GPRS.
     020214  004736                              JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
2588 020216  000207                              RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                       - CNTERR -
                                ;** ************************************************************************
                                ;*                                - COUNT ERROR ROUTINE -
                                ;*      THIS SUBROUTINE IS USED TO COUNT A "DATA" ERROR ON THE SPECIFIED
                                ;*      LINE.  IT CHECKS WHETHER ERROR SUMMARY REPORTING IS ACTIVE, OR SHOULD
                                ;*      BE MADE ACTIVE ON THIS LINE, AND ACTIVATES IT IF NECESSARY.
                                ;*
                                ;* INPUTS:      R5 - LINE NUMBER OF LINE UNDER CONSIDERATION.
                                ;*              ERCNTB - LABEL AT BASE OF ERROR COUNTERS TABLE.
                                ;*              ERSMRF - ERROR SUMMARY FLAGS (BIT SET IF LINE IN SUMMARY MODE).
                                ;*              NDERPT - NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE.
                                ;*
                                ;* OUTPUTS:     CARRY - SET IF LINE IS IN ERROR SUMMARY MODE.
                                ;*              ERCNT - ERROR COUNTER INCREMENTED FOR SPECIFIED LINE.
                                ;*              ERSMRF - BIT SET IF LINE SHOULD BE IN SUMMARY MODE.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,CNTERR
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

020220                          CNTERR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020220  004567  163552                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;*
                                ; COUNT THE ERROR ON THE COUNTER FOR THE SPECIFIED LINE.
                                ;-
020224  006305                          ASL     R5                      ;FORM WORD OFFSET FROM LINE NUMBER.
020226  016501  002464                  MOV     ERCNTB(R5),R1           ;GET THE PRESENT ERROR COUNT FOR THIS LINE.
020232  005201                          INC     R1                      ;COUNT ERROR.
020234  103402                          BCS     2$                      ;OVERFLOW?  YES, DON'T UPDATE COUNTER IN TABLE.
020236  010165  002464                   MOV     R1,ERCNTB(R5)          ;UPDATE ERROR COUNTER TABLE ENTRY.
020242  005767  161762          2$:      TST     NDERPT
020246  001411                          BEQ     60$                     ;SUMMARYS DISABLED?  YES, EXIT WITH CARRY 0.
020250  020167  161754                   CMP     R1,NDERPT              ;NO,  CHECK FOR ENOUGH ERRORS FOR SUMMARY USE.
020254  101002                          BHI     4$                      ;ENOUGH ERRORS TO USE SUMMARY?  YES, GO HANDLE.
020256  000241                           CLC                            ;INDICATE NOT TO USE SUMMARY REPORT YET.
020260  000404                          BR      60$                     ;EXIT WITH CARRY 0.
020262  056567  002410  162172  4$:      BIS     BITTBL(R5),ERSMRF       ;SET THE ERROR SUMMARY FLAG FOR LINE.
020270  000261                           SEC                            ;INDICATE TO USE SUMMARY REPORT.
020272                          60$:     PASS                           ;RESTORE GPRS.
020272  004736                                   JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020274  000207                          RTS     PC
```

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

020276                          DELAY:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
020276  004567  163474                        JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
020302  010401                        MOV     R4,R1           ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
020304  012702  177777                MOV     #-1,R2          ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
020310  005003                        CLR     R3              ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
020312  012704  020334                MOV     #62$,R4         ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
020316  004767  000130                JSR     PC,MSLOOP       ;DELAY THE REQUESTED # OF MS.
020322  103002                        BCC     60$             ;EXIT ROUTINE IF WE TIMED-OUT.]
020324  004767  000136                JSR     PC,OOPS         ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
020330                          60$:    PASS                    ;RESTORE GPRS.
020330  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
020332  000207                        RTS     PC

020334  177777                  62$:    .WORD   -1              ;DUMMY, NON-ZERO WORD.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - MSLGET -
                        ;**************************************************************************
                        ;*      - MILLI SECONDS LOOP WHICH RETURNS READ WORD AND REMAINING TIME -
                        ;*      THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE.  IT IS USED
                        ;*      TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD.  THE
                        ;*      CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
                        ;*      DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
                        ;*      THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
                        ;*      ROUTINE AND THEN ONCE EACH MILLI-SECOND THERE AFTER.
                        ;*      UPON RETURN, THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION
                        ;*      IS RETURNED BY THIS SUBROUTINE.
                        ;*
                        ;* INPUTS:      R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
                        ;*              R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
                        ;*              R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
                        ;*              R4 - ADDRESS OF THE WORD TO TEST.
                        ;*              MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
                        ;*
                        ;* OUTPUTS:     R0 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
                        ;*              R1 - REMAINING NUMBER OF MS IN TIME-OUT TIME.
                        ;*              CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,MSLGET
                        ;*
                        ;* COMMENTS:    THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
                        ;*              CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
                        ;*              ON THE SYSTEM.
                        ;*              THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
                        ;*              DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
                        ;*              LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
                        ;*              IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
                        ;*              THE DESIRED CONDITION BEFORE RETURNING.  IT INDICATES SUCCESS
                        ;*              IF THE CONDITION IS MET, FAILURE OTHERWISE.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;**************************************************************************

020336                  MSLGET:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020336  004567  163434                  JSR     R5,PREGOS       ;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.
                        ;-
020342  005102                  COM     R2              ;GET MASK OF UNUSED BITS.
020344  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.
                        ;-
020346  005701                  TST     R1              ;TEST THE TIME-OUT VALUE FOR ZERO.
020350  001011                  BNE     2$              ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
020352  011400                  MOV     (R4),R0         ;GET THE WORD TO TEST BEFORE EXITING.
020354  010067  000070          MOV     R0,62$          ;SAVE VALUE SO WE CAN RETURN IT.
020360  040200                  BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
020362  020003                  CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
020364  000261                  SEC                     ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
020366  001420                  BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
```

GLOBAL SUBROUTINE - MSLGET -

```
2720 020370  000241                    CLC                     ;INDICATE FAILURE (TIME-OUT).
2721 020372  000416                    BR      6$              ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
2722                                ;+
2723                                ; NON-ZERO TIME-OUT VALUE.  LOOP, WAITING FOR CONDITION OR TIME-OUT.
2724                                ;-
2725 020374  016705  161776        2$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2726 020400  011400                4$:     MOV     (R4),R0         ;GET THE WORD TO TEST.
2727 020402  010067  000042                MOV     R0,62$          ;SAVE WORD IN CASE THIS IS THE LAST.
2728 020406  040200                        BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
2729 020410  020003                        CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
2730 020412  000261                        SEC                     ;SET CARRY IN CASE OF SUCCESS.
2731 020414  001405                        BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
2732 020416  005305                        DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2733 020420  001367                        BNE     4$              ;LOOP IF MS NOT UP.
2734 020422  005301                        DEC     R1              ;DECREMENT THE MS TIME COUNT.
2735 020424  001363                        BNE     2$              ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
2736 020426  000241                        CLC                     ;CLEAR CARRY, WE TIMED-OUT.
2737                                ;+
2738                                ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME-OUT VALUE).
2739                                ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD.  EXIT ROUTINE.
2740                                ;-
2741 020430  016700  000014        6$:     MOV     62$,R0          ;PASS OUT THE LAST READ WORD.
2742 020434                        60$:    PASS    R0,R1           ;RESTORE GPRS, EXCEPT THE FOLLOWING:
     020434  010066  000002                             MOV     R0,ROSLOT(SP)       ;PUT R0 IN STACK SLOT.
     020440  010166  000004                             MOV     R1,R1SLOT(SP)       ;PUT R1 IN STACK SLOT.
     020444  004736                                     JSR     PC,@(SP)+           ;RETURN TO PREGO5 SUBRT.
2743                                                                ;R0 - LAST READ WORD CHECKED FOR CONDITION.
2744                                                                ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
2745 020446  000207                        RTS     PC              ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
2746                                ;+
2747                                ; LOCAL STORAGE.
2748                                ;-
2749 020450  000000                62$:    .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.
                        ;*******************************************************************************

020452                  MSLOOP:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
020452  004567  163320                  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.
                        ;-
020456  004767  177654          JSR     PC,MSLGET       ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.

020462                  60$:    PASS                    ;RESTORE GPRS,
020462  004736                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020464  000207          RTS     PC                ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
```

```
                                .SBTTL  GLOBAL SUBROUTINE                       - OOPS -
                                ;++ ************************************************************************
                                ;*                      - PROGRAM ABORT SUBROUTINE -
                                ;*      THIS SUBROUTINE IS USED TO ABORT THE PROGRAM WHEN A FATAL ERROR IS
                                ;*      DETECTED IN THE PROGRAM OR THE HOST SYSTEM HARDWARE.  AN ERROR MESSAGE
                                ;*      IS PRINTED GIVING SOME INFORMATION ABOUT THE NATURE OF THE ABORT.
                                ;*
                                ;* INPUTS:      R1 - ERROR CODE GIVING REASON FOR ABORT.
                                ;*
                                ;* OUTPUTS:     AN ERROR MESSAGE IS PRINTED.
                                ;*              A LIST OF RETURN PC VALUES FOR ALL SUBROUTINE CALLS IS PRINTED.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,OOPS
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

020466                          OOPS::  SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
020466  004567  163304                        JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
                                    ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
020472                                  ERRSF   101,EM0101
020472  104454                                                                  TRAP    C$ERSF
020474  000145                                                                  .WORD   101
020476  020532                                                                  .WORD   EM0101
020500  000000                                                                  .WORD   0
                                    ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
020502                                  PRINTF  @EM0102
020502  012746  020616                                                          MOV     @EM0102,-(SP)
020506  012746  000001                                                          MOV     @1,-(SP)
020512  010600                                                                  MOV     SP,R0
020514  104417                                                                  TRAP    C$PNTF
020516  062706  000004                                                          ADD     @4,SP
020522                          2$:     BREAK                           ;LOOK FOR OPERATOR CONTROL-C INPUT.
020522  104422                                                                  TRAP    C$BRK
020524  000776                          BR      2$                      ;INFINITE LOOP.
020526                          60$:    PASS                            ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
020526  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
020530  000207                          RTS     PC                      ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.

020532  110  117  123  EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
020535  124  040  103
020540  117  115  120
020543  125  124  105
020546  122  040  110
020551  101  122  104
020554  127  101  122
020557  105  040  117
020562  122  040  123
020565  117  106  124
020570  127  101  122
020573  105  040  102
020576  125  107  040
020601  105  116  103
020604  117  125  116
020607  124  105  122
```

GLOBAL SUBROUTINE - OOPS -

```
        020612  105     104     056
        020615  000
  2825  020616  045     116     045     EM0102:: .ASCIZ /%N%APROGRAM HUNG, WAITING FOR A CONTROL-C. <%%%%%%%%%%%%%%%%N%N/
        020621  101     120     122
        020624  117     107     122
        020627  101     115     040
        020632  110     125     116
        020635  107     054     040
        020640  127     101     111
        020643  124     111     116
        020646  107     040     106
        020651  117     122     040
        020654  101     040     103
        020657  117     116     124
        020662  122     117     114
        020665  055     103     056
        020670  040     074     052
        020673  052     052     052
        020676  052     052     052
        020701  052     052     052
        020704  052     052     052
        020707  045     116     045
        020712  116     000
  2826                                  .EVEN
```

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

020714                          PUFIFO::SAVE                            ;SAVE CONTENTS OF GPRS RO THRU R5.
020714  004567  163056                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020720  012701  001000                  MOV     #512.,R1                ;SET MAXIMUM TRY COUNT OF 512.
020724  016704  161316                  MOV     RBUFA,R4                ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.

020730  011402                  2$:     MOV     (R4),R2                 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
020732  100016                          BPL     6$                      ;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.
                                ;-
020734  012700  070000                  MOV     #70000,R0               ;GENERATE A BIT MAP OF CHAR ERROR BITS
020740  040200                          BIC     R2,R0                   ; WHICH ARE NOT SET FOR CHAR.
020742  001006                          BNE     4$                      ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
                                ;+
                                ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
                                ;-
020744  012700  000300                  MOV     #300,R0                 ; CHECK IF BMP OR SELFTEST?.
020750  040200                          BIC     R2,R0                   ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
020752  001002                          BNE     4$                      ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
020754  004767  001306                  JSR     PC,SAVBMP               ;SAVE BMP CODE ON THE QUEUE.

020760  005301                  4$:     DEC     R1                      ;DECREMENT THE TRY COUNT.
020762  001362                          BNE     2$                      ;LOOP TO TRY AGAIN.
020764  000241                          CLC                             ;CLEAR CARRY,TO INDICATE FIFO NOT PURGED.
020766  000401                          BR      60$                     ;EXIT WITH CARRY CLEAR.
020770  000261                  6$:     SEC                             ;SET CARRY, TO INDICATE FIFO PURGED.

020772                          60$:    PASS                            ;RESTORE GPRS,
020772  004736                                          JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
                                                                        ;CARRY BIT, SET INDICATES FIFO PURGED.
020774  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                       - RDPDR -
                        ;**************************************************************************
                        ;*      - READ AND VERIFY DATA PATTERN FROM DEVICE REGISTERS ROUTINE -
                        ;*      THIS ROUTINE READS AND VERIFIES THE ROTATED DATA PATTERN WHICH HAS
                        ;*      BEEN WRITTEN BY THE WDPDR SUBROUTINE.
                        ;*      EACH ACTIVE LINE'S REGISTER'S CONTENTS IS READ AND COMPARED WITH THE
                        ;*      WRITTEN DATA.
                        ;*      AFTER THE UNUSED AND READ ONLY (RO) BITS ARE MASKED OUT, ANY ERRORS ARE
                        ;*      REPORTED FROM THIS ROUTINE.
                        ;*      THIS ROUTINE WILL TAKE INTO ACCOUNT THE TYPE OF WRITE OPERATION WHICH
                        ;*      WAS PERFORMED BY THE WDPDR SUBROUTINE.
                        ;*
                        ;* INPUTS:      R2 - USED TO PASS IN THE DATA PATTERN TO BE ROTATED & VERIFIED.
                        ;*              R3 - BYTE INDICATOR (- => LO BYTE, + => HI BYTE, 0 => BOTH).
                        ;*              R4 - OPERATION TYPE INDICATOR (- => BIC, + => BIS, 0 => MOV).
                        ;*              ACTLNS - BIT MAP OF ACTIVE LINES ON THE DEVICE UNDER TEST.
                        ;*              CSRA - CONTAINS THE CSR ADDRESS OF THE DEVICE UNDER TEST.
                        ;*              DRADRT - BASE ADDRESS OF DEVICE REGISTER ADDRESS TABLE.
                        ;*              ERCNTB - LABEL AT BASE OF ERROR COUNTERS TABLE FOR LINES.
                        ;*              ERRMSG - SET UP WITH THE PROPER ERROR MESSAGE FOR THIS TEST.
                        ;*              ERRNBR - SET UP WITH THE PROPER ERROR NUMBER.
                        ;*              LPR0 - EQUATED TO LPR REG OFFSET FROM DEVICE CSR ADDRESS.
                        ;*              NUMLNS - NUMBER OF LINES ON THE DEVICE UNDER TEST.
                        ;*              NDERPT - NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE.
                        ;*              TXBFC0 - EQUATED TO TBUFFCT REG OFFSET FROM DEVICE CSR ADDRESS.
                        ;*              UNBTTB - BASE ADDRESS OF THE UNUSED BIT TABLE.
                        ;*
                        ;* OUTPUTS:     ERROR MESSAGES MAY BE PRINTED AT THE OPERATOR'S CONSOLE.
                        ;*              ERCNT - ERROR COUNTERS TABLE IS UPDATED FOR LINE UNDER TEST.
                        ;*              ERRBLK - CONTENTS DESTROYED.
                        ;*              ERSMRF - ERROR SUMMARY FLAGS BIT SET IF LINE IN SUMMARY MODE.
                        ;*              UUT CSR - ALL BITS CLEARED, EXCEPT IND.ADR.REG FIELD DESTROYED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RDPDR
                        ;*
                        ;* COMMENTS:    FOR BYTE ACCESSES, ONLY THE SPECIFIED BYTE IS VERIFIED.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ER1601,ROLDAP.
                        ;-- **************************************************************************

020776                  RDPDR:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
020776  004567  162774                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021002  012767  016710  162764          MOV     @ER1601,ERRBLK  ;SET UP THE ADDRESS OF THE ERROR REPORT RTN.
                        ;*
                        ; DETERMINE WHETHER REGISTER DATA SHOULD BE INVERTED FROM DATA PATTERN.
                        ;-
021010  005704                 TST     R4              ;CHECK THE OPERAND TYPE INDICATOR.
021012  100001                BPL     2$              ;BIC WRITE PERFORMED?  NO, USE STANDARD DATA.
021014  005102                 COM     R2              ;YES, INVERT THE DATA PATTERN.
                        ;*
                        ; SET UP OUTER LOOP.
                        ;-
021016  005005         2$:     CLR     R5              ;CLEAR LINE COUNTER TO SELECT LINE 0.
                        ;*
                        ; THE OUTER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP READS AND COMPARES DATA
                        ; FROM ALL OF THE DEVICE REGISTERS FOR A PARTICULAR LINE IF THE LINE IS ACTIVE.
                        ;-
```

```
 2934 021020  010267  000170          4$:     MOV     R2,70$          ;SAVE THE OUTER LOOP DATA PATTERN.
 2935 021024  010577  161214                  MOV     R5,@CSRA        ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
 2936 021030  010500                          MOV     R5,R0
 2937 021032  006300                          ASL     R0
 2938 021034  036067  002410  161174          BIT     BITTBL(R0),ACTLNS
 2939 021042  001452                        BEQ     16$             ;IS THE LINE ACTIVE?  NO, SKIP THE LINE.
 2940 021044  012703  000004                  MOV     #LPR0,R3        ;YES, INITIALIZE REGISTER OFFSET FOR LPR.
 2941                                  ;+
 2942                                  ;       THE INNER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP READS AND COMPARES
 2943                                  ;       DATA FROM A DEVICE REGISTER.
 2944                                  ;-
 2945 021050  010204                  6$:     MOV     R2,R4           ;SAVE THE INNER LOOP DATA PATTERN.
 2946 021052  046302  002264                  BIC     UNBTTB(R3),R2   ;REMOVE UNUSED BITS FROM EXPECTED DATA.
 2947 021056  016300  002244                  MOV     DRADRT(R3),R0
 2948 021062  005766  000010                  TST     R3SLOT(SP)      ;CHECK THE ACCESS TYPE INDICATOR.
 2949 021066  001002                        BNE     8$              ;BYTE ACCESS?  YES, GO PERFORM BYTE READ.
 2950 021070  011001                          MOV     (R0),R1         ;NO, PERFORM WORD READ OF DEVICE REGISTER.
 2951 021072  000416                        BR      12$
 2952 021074  100410                  8$:     BMI     10$             ;LOW BYTE ACCESS?  YES, GO DO LOW BYTE READ.
 2953 021076  005200                          INC     R0              ;HIGH BYTE ACCESS.  FORM HIGH BYTE ADDRESS.
 2954 021100  111001                          MOVB    (R0),R1         ;READ THE HI BYTE OF THE DUT REGISTER.
 2955 021102  000301                          SWAB    R1              ;PUT HI BYTE BACK INTO THE HI BYTE.
 2956 021104  042701  000377                  BIC     #377,R1         ;REMOVE THE UNUSED BYTE IN ACTUAL DATA.
 2957 021110  042702  000377                  BIC     #377,R2         ;REMOVE THE UNUSED BYTE IN EXPECTED DATA.
 2958 021114  000405                        BR      12$
 2959 021116  111001                  10$:    MOVB    (R0),R1         ;READ THE LOW BYTE OF THE DUT REGISTER.
 2960 021120  042701  177400                  BIC     #177400,R1      ;REMOVE THE UNUSED BYTE.
 2961 021124  042702  177400                  BIC     #177400,R2      ;FORM EXPECTED LOW BYTE FOR COMAPARISON.
 2962
 2963 021130  046301  002264          12$:    BIC     UNBTTB(R3),R1   ;REMOVE UNUSED BITS FROM ACTUAL DATA.
 2964 021134  020102                          CMP     R1,R2           ;COMPARE ACTUAL AND EXPECTED DATA.
 2965 021136  001404                        BEQ     14$             ;ACTUAL = EXPECTED?  YES, SKIP ERROR.
 2966 021140  004767  177054                JSR     PC,CNTERR       ;NO, COUNT THE ERROR, CHECK FOR ERROR SUMMARY.
 2967 021144  103401                        BCS     14$             ;USE ERROR SUMMARY?  YES, SKIP ERROR.
 2968                                  ;NO, REPORT "BAD BIT(S) IN DEVICE XXXXX REGISTER FOR LINE NN (D)."
 2969 021146                            ERROR
      021146  104460                                                                  TRAP    C$ERROR
 2970 021150  010402                  14$:    MOV     R4,R2           ;RESTORE THE INNER LOOP DATA PATTERN.
 2971 021152  004767  000410                  JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
 2972 021156  062703  000002                  ADD     #2,R3           ;SET REGISTER OFFSET TO THE NEXT REGISTER.
 2973 021162  020327  000016                  CMP     R3,#TXBFC0      ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
 2974 021166  003730                        BLE     6$              ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
 2975                                  ;+
 2976                                  ; BACK INTO THE OUTER LOOP.  NOW SET UP FOR NEXT LINE.  LOOP IF NOT DONE.
 2977                                  ;-
 2978 021170  016702  000020          16$:    MOV     70$,R2          ;SET UP TO ROTATE THE DATA PATTERN.
 2979 021174  004767  000366                  JSR     PC,ROLDAP       ;ROTATE THE DATA PATTERN.
 2980 021200  005205                          INC     R5              ;COUNT THIS LINE
 2981 021202  020527  000010                  CMP     R5,#NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
 2982 021206  002704                        BLT     4$              ;LOOP IF SOME LINES NOT DONE.
 2983
 2984 021210                          60$:    PASS                    ;RESTORE GPRS.
      021210  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
 2985 021212  000207                        RTS     PC
 2986
 2987 021214  000000                  70$:    .WORD   0               ;STORAGE FOR DATA PATTERN OUTSIDE INNER LOOP.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - REGTST -
                        ;** ************************************************************************
                        ;*                          - REGISTERS TEST SUBROUTINE -
                        ;*      SUBROUTINE TO TEST THE DEVICE UNDER TEST (DUT) REGISTERS.  THE USED
                        ;*      BITS OF THE REGISTERS ARE EITHER ALL CLEARED OR ALL SET AND THEN THE
                        ;*      DATA PATTERN IS WRITTEN AND VERIFIED USING EITHER WORD OR BYTE
                        ;*      ACCESSES IN READ/WRITE OR READ/MODIFY/WRITE MODE.
                        ;*
                        ;* INPUTS:      R3 - BYTE INDICATOR (- => LOW, + => HIGH, 0 => BOTH BYTES).
                        ;*              R4 - ACCESS MODE (-1 => SET THEN BIC, 1 => CLEAR THEN BIS,
                        ;*                   (-2 => SET THEN MOV, +2 CLEAR THEN MOV).
                        ;*              ERRNBR - SET UP WITH INITIAL ERROR NUMBER.
                        ;*
                        ;* OUTPUTS:     GPRS0 - GPR SAVE AREA 0 IS DESTROYED.
                        ;*              DEVICE UNDER TEST REGISTERS ARE WRITTEN.
                        ;*              ERROR MESSAGES MAY BE PRINTED AT THE OPERATORS CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,REGTST
                        ;*
                        ;* COMMENTS:    THIS ROUTINE LOOP 16 TIMES WRITING THE SAME DATA PATTERN
                        ;*              ROTATED LEFT ONCE EACH ITERATION.
                        ;*              THIS ROUTINE CAN REPORT ERRORS INITIAL ERRNBR THRU INITIAL+2.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: RDPDR,ROLDAP,SWAPO,WDPDR
                        ;-- ************************************************************************

021216                  REGTST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021216  004567  162554                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; SET UP THE GPRS FOR THE WRITTING OF THE DATA PATTERN.
                        ;-
021222  012705  000020                  MOV     #16.,R5         ;SET UP LOOP COUNTER TO COUNT 16 ITERATIONS.
021226  012702  167410                  MOV     #167410,R2      ;INITIALIZE THE DATA PATTERN.
021232  032704  000001                  BIT     #BIT0,R4        ;TEST FOR R/W ACCESS.
021236  001001                          BNE     2$              ;R/M/W ACCESS?  YES, R4 IS ALL SET UP.
021240  005004                          CLR     R4              ;NO, INDICATE R/W ACCESS.
021242                          2$:
                        ;*
                        ; SET UP THE GPRS FOR THE CLEARING OR SETTING OF ALL THE USED BITS.
                        ;-
021242  010400                          MOV     R4,R0           ;PASS OPERATION TYPE INDICATOR AROUND SWAPO.
021244  004767  001142                  JSR     PC,SWAPO        ;GET ALTERNATE GPR SET IN R1 THRU R5.
021250  016701  162514                  MOV     ERRNBR,R1       ;SAVE THE INITIAL ERROR NUMBER.
021254  010004                          MOV     R0,R4
021256  005404                          NEG     R4              ;SET UP OP TYPE FOR CLEARING OR SETTING.
021260  005002                          CLR     R2              ;SET UP CLEAR WRITE PATTERN.
021262  026627  000012  000002          CMP     R4SLOT(SP),#2   ;TEST FOR CLEAR THEN MOV TEST SEQUENCE.
021270  001401                          BEQ     4$              ;CLEAR THEN MOV?  YES, LEAVE WRITE PAT CLEAR.
021272  005102                          COM     R2              ;NO, SET ALL BITS OF WRITE PATTERN.
021274  005003                  4$:     CLR     R3              ;INDICATE THAT WORD ACCESSES SHOULD BE USED.
021276  005000                          CLR     R0              ;SET ALTERNATE BYTE EXPECTED DATA PAT TO CLEAR.
021300  026627  000012  177776          CMP     R4SLOT(SP),#-2  ;TEST FOR SET THEN MOV TEST SEQUENCE.
021306  001001                          BNE     6$              ;SET THEN MOV?  YES, LEAVE ALT BYTE PAT CLEAR.
021310  005100                          COM     R0              ;NO, SET ALT BYTE EXPECTED DATA PAT TO ALL 1'S.
021312  004767  001074          6$:     JSR     PC,SWAPO        ;RESTORE SWAPPED GPR VALUES TO R1 THRU R5.
                        ;*
                        ; START OF DATA PATTERN LOOP.
```

GLOBAL SUBROUTINE - REGTST -

```
3045                                    ;-
3046 021316                         8$:
3047                                    ;+
3048                                    ; SET OR CLEAR ALL THE USED BITS OF THE DEVICE REGISTERS FOR ALL LINES.
3049                                    ; VERIFY THAT ALL THE BITS WERE SET OR CLEARED CORRECTLY.
3050                                    ;-
3051 021316  004767  001070                JSR     PC,SWAPO        ;GET ALTERNATE GPRS FOR SETTING INTIAL STATES.
3052 021322  004767  002006                JSR     PC,WDPDR        ;GO CLEAR ALL USED REGISTER BITS, ALL LINES.
3053 021326  010167  162436                 MOV    R1,ERRNBR       ;SET UP ERROR NUMBER TO INITIAL ERRNBR.
3054 021332  004767  177440                JSR     PC,RDPDR        ;VERIFY ALL USED REGISTER BITS, ALL LINES.
3055 021336  004767  001050                JSR     PC,SWAPO        ;RESTORE MAIN GPRS CONTENTS.
3056                                    ;+
3057                                    ; WRITE DATA PATTERNS, ALL LOWER BYTE USED BITS, ALL REGISTERS, ALL LINES.
3058                                    ; VERIFY THAT THE DATA PATTERN WAS WRITTEN CORRECTLY.
3059                                    ;-
3060 021342  004767  001766                JSR     PC,WDPDR        ;WRITE DATA PATTERN TO DEVICE REGISTERS.
3061 021346  005267  162416                 INC    ERRNBR          ;SET ERROR NUMBER TO INITIAL+1.
3062 021352  004767  177420                JSR     PC,RDPDR        ;VERIFY DATA PATTERN IN ALTERRED BYTE(S).
3063 021356  005703                        TST     R3              ;CHECK THE BYTE INDICATOR.
3064 021360  001411                        BEQ     10$             ;WORD ACCESS?  YES, SKIP SECOND BYTE CHECK.
3065                                    ;+
3066                                    ; CHECK THAT THE ALTERNATE (UNMODIFIED) BYTE IS CLEAR OR SET AS EXPECTED.
3067                                    ;-
3068 021362  010201                         MOV    R2,R1           ;SAVE THE DATA PATTERN.
3069 021364  010002                         MOV    R0,R2           ;GET THE ALTERNATE BYTE EXPECTED DATA.
3070 021366  005403                         NEG    R3              ;INDICATE THAT OTHER BYTE IS TO BE CHECKED.
3071 021370  005267  162374                 INC    ERRNBR          ;SET ERROR NUMBER TO INITIAL+2.
3072 021374  004767  177376                JSR     PC,RDPDR        ;VERIFY DATA PATS IN OTHER BYTES OF REGISTERS.
3073 021400  005403                         NEG    R3              ;RESTORE BYTE INDICATOR.
3074 021402  010102                         MOV    R1,R2           ;RESTORE DATA PATTERN.
3075                                    ;+
3076                                    ; PEPARE THE NEXT DATA PATTERN AND LOOP IF NOT DONE.
3077                                    ;-
3078 021404  004767  000156        10$:     JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
3079 021410  005305                         DEC    R5              ;COUNT THIS ITERATION OF THE LOOP.
3080 021412  003341                        BGT     8$              ;ALL PATTERNS DONE?  NO, LOOP.
3081                                                                ;YES, RESTORE ERROR NUMBER AND EXIT.
3082 021414  016767  161030  162346  60$:    MOV     GPRSOB,ERRNBR   ;GET THE ERROR NUMBR FROM GPR SWAP STORAGE.
3083 021422                                 PASS                    ;RESTORE GPRS.
     021422  004736                                     JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
3084 021424  000207                        RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - REPSMR -
                        ;** ************************************************************************
                        ;*                        - REPORT ERROR SUMMARY ROUTINE -
                        ;*      THIS SUBROUTINE REPORTS AN ERROR SUMMARY FOR THOSE LINES WHICH HAVE
                        ;*      EXCEEDED THE NUMBER OF INDIVIDUAL ERRORS TO REPORT FOR A SINGLE LINE
                        ;*      IN A SINGLE TEST.  THIS PARAMETER CAN BE SPECIFIED BY THE OPERATOR IF
                        ;*      HE/SHE ANSWERS THE SOFTWARE PARAMETER QUESTIONS.
                        ;*
                        ;* INPUTS:      ERCNTB - LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
                        ;*              ERRMSG - ADDRESS OF PRIMARY ERROR MESSAGE.
                        ;*              ERRNBR - ERROR NUMBER OF ERRORS IN THIS ROUTINE.
                        ;*              ERSMRF - "REPORT ERROR SUMMARY FOR LINE" FLAGS.
                        ;*
                        ;* OUTPUTS:     ERRBLK - ADDRESS OF ERROR REPORTING ROUTINE (DESTROYED).
                        ;*              SUMMARY MESSAGES MAY BE PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,REPSMR
                        ;*
                        ;* COMMENTS:    IF NO LINES HAVE EXCEEDED THE MAXIMUM NUMBER OF INDIVIDUAL
                        ;*              ERRORS TO REPORT, NO MESSAGES ARE PRINTED BY THIS ROUTINE.
                        ;*              ERROR SUMMARIES IN THIS ROUTINE ARE REPORTED AS ERRORS.
                        ;*              THE CONTENTS OF ERRBLK ARE DESTROYED.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED:
                        ;-- ************************************************************************

021426                  REPSMR:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
021426  004567  162344          JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
021432  005767  161024          TST     ERSMRF          ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
021436  001404                  BEQ     60$             ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
                        ;*
                        ; WE HAVE SOME ERROR SUMMARIES TO REPORT.
                        ;-
021440  012767  017156  162326          MOV     #ER9004,ERRBLK  ;SELECT ERROR REPORTING ROUTINE.
                        ;*
                        ; REPORT
                        ;     "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
                        ;-
021446                          ERROR
021446  104460                                                  TRAP    C$ERROR

021450                  60$:    PASS                    ;RESTORE GPRS.
021450  004736                          JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
021452  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.
                        ;*************************************************************************

021454                  RESETT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
021454  004567  162316          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021460  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.
                        ;-
021464  016704  160554          MOV     CSRA,R4         ;GET THE ADDRESS OF THE DUT'S CSR.
021470  030214                  BIT     R2,(R4)         ;CHECK STATE OF MASTER RESET BIT.
021472  001406                  BEQ     2$              ;DON'T DELAY IF MR IS ALREADY CLEAR.
021474  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
021476  012701  004704          MOV     #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
021502  004767  176630          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
021506  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.
                        ;-
021510  010277  160530  2$:     MOV     R2,@CSRA        ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
021514  004767  000614          JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
                          ;*
                          ;   SET SELF-TEST TIME-OUT OF 2.5 SECONDS, AND WAIT FOR M.R TO CLEAR.
                          ;   IF TIME-OUT OCCURS, THEN REPORT THE FATAL ERROR AND PASS-OUT THE ABORT
                          ;   TEST INDICATOR.
                          ;-
021520  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
021522  012701  004704          MOV     #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
021526  004767  176604          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
021532  103410                  BCS     6$              ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
```

GLOBAL SUBROUTINE - RESETT -

```
3184                                    ;*
3185                                    ; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET,TEST ABORTED".
3186                                    ; INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
3187                                    ;-
3188 021534  012701  012360            4$:     MOV     #EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
3189 021540  012767  017006  162226            MOV     #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
3190                                            ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
3191                                            ; "TEST ABORTED"
3192 021546                                    ERROR                   ;                  >>>>> ERROR <<<<<
     021546  104460                                                                              TRAP    C$ERROR
3193 021550  000241                            CLC                     ;INDICATE TEST IS TO BE ABORTED.
3194 021552  000403                            BR      60$             ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
3195                                    ;*
3196                                    ; CLEAR TX AND RX INTERRUPT ENABLE STATUS FLAGS IN IESTAT.
3197                                    ; EXIT WITH CONTINUE TEST INDICATOR SET (IE,CARRY SET).
3198                                    ;-
3199 021554  005067  160550            6$:      CLR    IESTAT          ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
3200 021560  000261                             SEC                    ;INDICATE SUCCESS, CONTINUE TEST.
3201
3202 021562                            60$:    PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
     021562  004736                                    JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
3203                                                                    ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
3204 021564  000207                            RTS     PC
3205
```

GLOBAL SUBROUTINE - ROLDAP -

```
                        .SBTTL  GLOBAL SUBROUTINE                - ROLDAP -
                        ;*****************************************************************************
                        ;*      - ROTATE LEFT DATA PATTERN
                        ;*      THIS ROUTINE ROTATES THE PASSED INPUT DATA PATTERN LEFT,WITHOUT GOING
                        ;*      THROUGH THE CARRY.THE CARRY IS INITIALLY SET OR CLEARED DEPENDING
                        ;*      UPON THE STATE OF THE MSB OF THE DATA PATTERN,BEFORE A ROL INSTRUCTION
                        ;*      IS EXECUTED.
                        ;*
                        ;* INPUTS:      R2 - CONTAINS THE DATA PATTERN TO BE ROTATED
                        ;*
                        ;* OUTPUTS:     R2 - CONTAINS THE ROTATED DATA PATTERN
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,ROLDAP
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE
                        ;*****************************************************************************

021566                  ROLDAP::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
021566 004567 162204            JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
021572 010202           MOV     R2,R2                   ;SET PROCESSOR STATUS CODES
021574 005702           TST     R2                      ;CHECK MSB
021576 100402           BMI     2$                      ;BRANCH IF SET
021600 000241           CLC                             ;CLEAR CARRY BIT IF MSB CLEAR
021602 000401           BR      4$                      ;
021604 000261   2$:     SEC                             ;SET CARRY IF MSB SET
021606 006102   4$:     ROL     R2                      ;ROTATE DATA PATTERN LEFT
021610          60$:    PASS    R2                      ;RESTORE GPRS,EXCEPT
021610 010266 000006            MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
021614 004736                   JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
                        ;R2 - CONTAINS THE ROTATED DATA PATTERN

021616 000207           RTS     PC
```

GLOBAL SUBROUTINE - RSTRPT -

```
                        .SBTTL  GLOBAL SUBROUTINE                - RSTRPT -
                        ;++ ************************************************************************
                        ;*                        - REPORT ANY RESET ERRORS ROUTINE -
                        ;*      THIS ROUTINE DETERMINES IF ANY ERROR CODES ARE AMONG THE DIAGNOSTIC
                        ;*      CODES REPORTED PLACED IN THE DUT RECEIVED CHARACTER FIFO BY THE
                        ;*      SELF-TEST.  IF ANY NON BMP ERROR CODES ARE FOUND, OR IF OTHER ERRORS
                        ;*      ARE ENCOUNTERED, APPROPRIATE ERRORS ARE REPORTED.  ANY BMP CODES THAT
                        ;*      ARE FOUND, ARE PLACED ON THE BMP CODE QUEUE TO BE REPORTED LATER.
                        ;*      THIS ROUTINE ALSO PURGES THE DUT FIFO LOOKING FOR ANY CHARACTERS
                        ;*      OR MODEM STATUS CODES.  IF ANY ARE FOUND, ERRORS ARE REPORTED.
                        ;*
                        ;* INPUTS:      ERRMSG - ADDRESS OF THE PRIMARY ERROR MESSAGE.
                        ;*              ERRNBR - ERROR NUMBER OF FIRST ERROR REPORTED BY THIS ROUITNE.
                        ;*              NUMLNS - EQUATED TO THE NUMBER OF LINE ON THE DUT.
                        ;*              RBUFA - CONTAINS ADDRESS OF THE DUT RECEIVER FIFO.
                        ;*
                        ;* OUTPUTS:     CARRY - SUCCESS FLAG (SET IF FIFO CLEARED SUCCESSFULLY).
                        ;*              ERRBLK - ADDRESS OF THE ERROR REPORT ROUTINE (DESTROYED).
                        ;*              ERROR MESSAGES CAN BE PRINTED AT THE OPERATORS CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RSTRPT
                        ;*
                        ;* COMMENTS:    THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERRNBR
                        ;*              THRU INITIAL ERRNBR+4.
                        ;*              THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ER0503,ER9007,ER9008,SAVBMP.
                        ;-- ************************************************************************
021620                  RSTRPT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021620  004567  162152                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; READ CORRECT NUMBER (NUMBER OF LINE ON DUT) OF CHARS FROM THE FIFO.
                        ; VERIFY THAT EACH CHAR IS A SELFTEST SUCCESS CODE.
                        ;-
021624  005003                  CLR     R3                      ;CLEAR THE CODE COUNTER.
021626  016705  162136          MOV     ERRNBR,R5               ;SAVE ERRNBR FOR RESTORATION LATER.
021632  017702  160410  2$:     MOV     @RBUFA,R2               ;READ A CHAR FROM THE DUT FIFO.
021636  100412                  BMI     4$                      ;SKIP ERROR IF DATA.VALID SET FOR CHAR.
                        ;*
                        ; WE EXPECT A SELFTEST CODE, BUT THIS FIFO SLOT IS EMPTY.
                        ;-
                            MOV     R5,ERRNBR               ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
                            MOV     #EM9018,R1              ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
021650  012767  017260  162116    MOV     #ER9007,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
                        ;*
                        ; REPORT ERROR WITH NUMBER INITIAL ERRNRB.
                        ;    "NO SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET."
                        ;-
021656                      ERROR                       ;               >>>>> ERROR <<<<<.
021656  104460                                                                    TRAP    C$ERROR
                        ;*
                        ; INIDICATE "SUCCESS" (BECAUSE FIFO IS PURGED), AND EXIT THIS ROUTINE.
                        ;-
021660  000261              SEC                         ;SET SUCCESS FLAG.
021662  000545              BR      60$                 ;EXIT ROUTINE.
                        ;*
```

```
3293                                    ; DETERMINE IF THIS IS NOT A SELFTEST CODE.
3294                                    ;-
3295 021664  012700  070001        4$:      MOV     #70001,R0       ;GENERATE BIT MAP OF ANY CLEAR ERROR BITS OR
3296 021670  040200                         BIC     R2,R0           ;  BIT 0 WHICH ARE CLEAR.
3297 021672  001033                        BNE     8$              ;GO TO REPORT ERROR IF THIS IS NOT A TEST CODE.
3298                                    ;+
3299                                    ; WE HAVE A TEST CODE (EITHER BMP OR SELFTEST CODE).
3300                                    ; DETERMINE WHAT TYPE OF CODE WE HAVE.
3301                                    ;-
3302 021674  032702  000200                 BIT     #BIT7,R2        ;TEST ROM VERSION CODE INDICATOR BIT.
3303 021700  001443                        BEQ     10$             ;SKIP ERRORS IF SELFTEST ROM VERSION CODE.
3304 021702  120227  000203                 CMPB    R2,#203         ;CHECK IF SKIP SELF TEST CODE.
3305 021706  001440                        BEQ     10$             ;SKIP ERROR REPORT IF SKIP SELF TEST CODE FOUND
3306 021710  120227  000201                 CMPB    R2,#201         ;CHECK IF NULL CODE PRESENT.
3307 021714  001435                        BEQ     10$             ;SKIP ERROR REPORT IF SELF TEST NULL CODE.
3308 021716  012700  000300                 MOV     #300,R0         ;TEST CODE TYPE BITS FOR BOTH CODE
3309 021722  040200                         BIC     R2,R0           ;  TYPE BITS SET (INDICATING BMP CODE).
3310 021724  001003                        BNE     6$              ;IF IT IS NOT A BMP CODE GO REPORT ERROR.
3311 021726  004767  000334                 JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
3312 021732  000426                        BR      10$             ;GO GET THE NEXT CHARACTER FROM THE FIFO.
3313                                    ;+
3314                                    ; WE HAVE A SELFTEST ERROR CODE.
3315                                    ;-
3316 021734  010567  162030        6$:      MOV     R5,ERRNBR       ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
3317 021740  005267  162024                 INC     ERRNBR          ;CALCULATE INITIAL ERROR NUMBER PLUS 1.
3318 021744  012701  015545                 MOV     #EM9020,R1      ;PASS ERROR MESSAGE INFO TO ER9008 ROUTINE.
3319 021750  012767  017336  162016         MOV     #ER9008,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3320                                    ;+
3321                                    ; REPORT ERROR WITH NUMBER INITIAL ERRNRB + 1.
3322                                    ;    "UNEXPECTED SELFTEST ERROR CODE FOR LINE NN IN FIFO AFTER RESET:"
3323                                    ;-
3324 021756                                 ERROR                   ;                  >>>>> ERROR <<<<<.
      021756  104460                                                                      TRAP    C$ERROR
3325 021760  000413                        BR      10$             ;GO TO END OF LOOP.
3326                                    ;+
3327                                    ; WE HAVE A NON-SELFTEST CODE (EITHER BMP CODE OR DATA CHAR).
3328                                    ;-
3329 021762  010567  162002        8$:      MOV     R5,ERRNBR       ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
3330 021766  062767  000002  161774         ADD     #2,ERRNBR       ;CALCULATE INITIAL ERROR NUMBER PLUS 2.
3331 021774  012701  015530                 MOV     #EM9019,R1      ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
3332 022000  012767  017260  161766         MOV     #ER9007,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3333                                    ;+
3334                                    ; REPORT ERROR WITH NUMBER INITIAL ERRNRB + 2.
3335                                    ;    "NON-SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET."
3336                                    ;-
3337 022006                                 ERROR                   ;                  >>>>> ERROR <<<<<.
      022006  104460                                                                      TRAP    C$ERROR
3338                                    ;+
3339                                    ; END OF LOOP, LOOP IF NOT ALL CHARS HAVE BEEN READ FROM THE FIFO.
3340                                    ;-
3341 022010  005203                10$:     INC     R3              ;SET CODE COUNTER FOR NEXT ITERATION OF LOOP.
3342 022012  020327  000010                 CMP     R3,#NUMLNS      ;TEST FOR ALL CODES READ.
3343 022016  002705                        BLT     2$              ;LOOP IF NOT CHARS READ FROM FIFO.
3344                                    ;+
3345                                    ; PURGE THE FIFO UNTIL DATA.VALID IS CLEAR OR UNTIL TOO MANY CHARS ARE READ.
3346                                    ;-
3347 022020  012704  000022                 MOV     #18.,R4         ;INITIALIZE THE CHARACTER COUNTER.
```

GLOBAL SUBROUTINE - RSTRPT -

```
3348 022024  010567  161740                    MOV     R5,ERRNBR       ;GET INITIAL VALUE OF THE ERROR NUMBER.
3349 022030  062767  000003  161732            ADD     #3,ERRNBR       ;CALCULATE ERROR NUMBER OF NEXT ERROR.
3350 022036  012767  017336  161730            MOV     #ER9008,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3351 022044  017702  160176          12$:      MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE DUT FIFO.
3352 022050  000261                            SEC                     ;INDICATE SUCCESS IN CASE DATA.VALID IS CLEAR.
3353 022052  100051                            BPL     60$             ;EXIT ROUTINE WITH SUCCESS IF DATA.VALID CLEAR.
3354                                    ;+
3355                                    ; WE HAVE A CHARACTER.
3356                                    ; DETERMINE IF CHARACTER IS A DATA CHARACTER.
3357                                    ;-
3358 022054  012700  070000                        MOV     #70000,R0       ;TEST BITS 12 THRU 14 OF THE
3359 022060  040200                            BIC     R2,R0           ;  CODE READ FROM THE DUT FIFO.
3360 022062  001403                            BEQ     14$             ;SKIP THIS ERROR IF CODE IS NOT A DATA CHAR.
3361                                    ;+
3362                                    ; WE HAVE AN UNEXPECTED DATA CHARACTER: SET UP AND GO TO REPORT ERROR.
3363                                    ;-
3364 022064  012701  015571                        MOV     #EM9022,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3365 022070  000423                            BR      22$             ;GO TO REPORT THIS ERROR.
3366                                    ;+
3367                                    ; WE HAVE AN UNEXPECTED CODE.
3368                                    ; DETERMINE IF THE CODE IS A MODEM STATUS CODE.
3369                                    ;-
3370 022072  032702  000001          14$:      BIT     #BIT0,R2        ;TEST MODEM STATUS INDICATOR BIT OF CODE.
3371 022076  001003                            BNE     16$             ;SKIP THIS ERROR IF NOT MODEM STATUS CODE.
3372                                    ;+
3373                                    ; WE HAVE A MODEM STATUS CODE: SET UP AND GO TO REPORT ERROR.
3374                                    ;-
3375 022100  012701  015610                        MOV     #EM9023,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3376 022104  000415                            BR      22$             ;GO TO REPORT THIS ERROR.
3377                                    ;+
3378                                    ; WE HAVE AN ONBOARD TEST CODE.
3379                                    ; DETERMINE IF THIS CODE IS A BMP CODE.
3380                                    ;-
3381 022106  032702  000200          16$:      BIT     #BIT7,R2        ;TEST THE ROM VERSION BIT OF THE CODE.
3382 022112  001404                            BEQ     18$             ;GOTO SET UP FOR SELFTEST CODE IF ROM VERSION.
3383 022114  012700  000300                        MOV     #300,R0
3384 022120  040200                            BIC     R2,R0           ;TEST THE ERROR TYPE BITS OF THE CODE.
3385 022122  001403                            BEQ     20$             ;SKIP THIS ERROR IF BMP CODE.
3386                                    ;+
3387                                    ; WE HAVE A SELFTEST CODE: SET UP AND GO TO REPORT ERROR.
3388                                    ;-
3389 022124  012701  015632          18$:      MOV     #EM9024,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3390 022130  000403                            BR      22$             ;GO TO REPORT THIS ERROR.
3391                                    ;+
3392                                    ; WE HAVE A BMP CODE: SAVE IT ON THE QUEUE.
3393                                    ;-
3394 022132  004767  000130          20$:      JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
3395 022136  000401                            BR      24$             ;
3396                                    ;+
3397                                    ; REPORT THE ERROR WITH ERROR NUMBER OF INITIAL ERRNBR + 3.
3398                                    ;    "UNEXPECTED XXX XXXX FOR LINE NN IN FIFO AFTER RESET:"
3399                                    ;-
3400 022140                            22$:      ERROR                   ;               >>>>> ERROR <<<<<.
      022140  104460                                                                      TRAP    C$ERROR
3401                                    ;+
3402                                    ; END OF LOOP.
3403                                    ; COUNT THE CHARACTER WE JUST RECEIVED, AND CHECK FOR TOO MANY RECEIVED.
```

GLOBAL SUBROUTINE - RSTRPT -

```
3404                                        ;-
3405 022142  005304                 24$:    DEC     R4              ;COUNT THIS CHARACTER.
3406 022144  001337                         BNE     12$             ;LOOP IF NOT TOO MANY CHARACTERS PURGED.
3407                                        ;+
3408                                        ; WE READ TOO MANY VALID CHARACTERS WHILE TRYING TO PURGE THE FIFO.
3409                                        ; REPORT ERROR AND EXIT WITHOUT SUCCESS.
3410                                        ;    "FIFO WILL NOT PURGE (DATA.VALID STUCK SET), REMAINDER OF TEST SKIPPED."
3411                                        ;-
3412 022146  012701  015407                 MOV     #EM9017,R1      ;SELECT PROPER ERROR MESSAGE.
3413 022152  010567  161612                 MOV     R5,ERRNBR       ;GET INITIAL ERROR NUMBER.
3414 022156  062767  000004  161604         ADD     #4,ERRNBR       ;CALCULATE INITIAL ERRNBR + 4.
3415 022164  012767  016466  161602         MOV     #ER0503,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3416                                        ;PRINT ERROR REPORT.
3417 022172                                 ERROR                   ;               >>>>> ERROR <<<<<.
     022172  104460                                                                    TRAP    C$ERROR
3418 022174  000241                         CLC                     ;CLEAR THE SUCCESS FLAG.
3419
3420 022176                         60$:    PASS                    ;RESTORE GPRS.
     022176  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3421 022200  000207                         RTS     PC              ; CARRY - SUCCESS FLAG (SET IF FIFO IS PURGED).
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - RXIEO -
                                ;++ ************************************************************************
                                ;*                          - 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,RXIEO
                                ;*
                                ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*               THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************
022202  010046                  RXIEO:: MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
022204                                  GETPRI -(SP)            ;SAVE PROCESSOR PRIORITY ON STACK.
022204  104440                                                                  TRAP    C$GPRI
022206  010046                                                                  MOV     R0,-(SP)
022210                                  SETPRI #PRI07           ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
022210  012700  000340                                                          MOV     #PRI07,R0
022214  104441                                                                  TRAP    C$SPRI
022216  042767  137777  160104        BIC     #137777,IESTAT  ;CLEAR RX.INT.ENBL BIT IN IESTAT.
022224  016777  160100  160012        MOV     IESTAT,@CSRA    ;DISABLE RX INTERRUPTS.
022232                                  SETPRI (SP)+            ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
022232  012600                                                                  MOV     (SP)+,R0
022234  104441                                                                  TRAP    C$SPRI
022236  012600                          MOV     (SP)+,R0        ;RESTORE R0.
022240  000207                  RTS     PC
```

GLOBAL SUBROUTINE - RXIE1 -

```
                        .SBTTL  GLOBAL SUBROUTINE                - RXIE1 -
                        ;** ************************************************************************
                        ;*                          - RECEIVER INTERRUPT ENABLE -
                        ;*      THIS ROUTINE IS USED TO ENABLE RECEIVER INTERRUPTS IN THE DHV11.
                        ;*
                        ;* INPUTS:       NONE.
                        ;*
                        ;* OUTPUTS:      THE RX.INT.ENBL BIT IS SET IN THE DUT CSR.
                        ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
                        ;*               ENABLE BITS.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RXIE1
                        ;*
                        ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                        ;*               THE DUT CSR ARE DESTROYED.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;- ************************************************************************

022242  052767  000100  160060  RXIE1::  BIS     #BIT06,IESTAT   ;SET RX.INT.ENBL BIT IN IESTAT.
022250  042767  137677  160052           BIC     #137677,IESTAT  ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
022256  016777  160046  157760           MOV     IESTAT,@CSRA    ;ENABLE RX INTERRUPTS.
022264  000207                           RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - SAVBMP -
                                ;** ************************************************************************
                                ;*                        - SAVE BMP CODES ROUTINE -
                                ;*      THIS ROUTINE SAVES THE PARAMETER PASSED IN, ONTO THE BMP CODE QUEUE
                                ;*      TOGETHER WITH THE NUMBER OF THE CURRENTLY EXECUTING TEST.
                                ;*
                                ;* INPUTS:      R2 - CONTAINS THE BMP CODE THAT IS TO BE PLACED ON THE QUEUE.
                                ;*              BMPCQP - CONTAINS ADDRESS OF NEXT LOCATION IN THE BMP QUEUE.
                                ;*              BMPCQB - LABEL AT BASE OF THE BMP CODE QUEUE.
                                ;*              BMPCQE - LABEL OF NEXT LOCATION AFTER THE END OF THE BMP QUEUE.
                                ;*              TSTNUM - CONTAINS THE NUMBER OF THE CURRENT TEST.
                                ;*
                                ;* OUTPUTS:     BMPCQP -  INCREMENTED BY 4.
                                ;*              THE CONTENTS OF THE BMP CODE QUEUE ARE UPDATED.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,SAVBMP
                                ;*
                                ;* COMMENTS:    IF THE OVERFLOW OCCURS THEN THE LAST LOCATION WILL BE
                                ;*              OVERWRITTEN BY ANY SUBSEQUENT ATTEMPTS TO UPDATE THE QUEUE.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

022266                          SAVBMP:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
022266  004567  161504                         JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
022272  016704  160226                 MOV     BMPCQP,R4       ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
022276  116724  160024                 MOVB    TSTNUM,(R4)+    ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
022302  005204                         INC     R4              ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
022304  042702  177400                 BIC     #177400,R2      ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
022310  010224                         MOV     R2,(R4)+        ;SAVE THE BMP CODE ON THE QUEUE.
022312  020427  002726                 CMP     R4,#BMPCQE      ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
022316  103402                         BLO     2$              ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
022320  162704  000004                 SUB     #4,R4           ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
022324  010467  160174         2$:     MOV     R4,BMPCQP       ;SAVE THE POINTER.

022330                          60$:    PASS                     ;RESTORE GPRS.
022330  004736                                         JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
022332  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.
                        ;-- ************************************************************************

022334                  SKPSTS:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
022334  004567  161436                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022340  012704  000012          MOV     #10.,R4         ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
022344  004767  175726          JSR     PC,DELAY        ;DELAY FOR 10 MILLI-SECONDS.
                        ;*
                        ; WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
                        ;-
022350  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 DHV-11.
022354  012703  052525          MOV     #52525,R3       ;INITIALISE THE SKIP SELF-TEST CODE.
022360  005301          4$:     DEC     R1              ;SELECT THE NEXT SET OF DEVICE REGISTERS.
022362  016704  157656          MOV     CSRA,R4         ;GET THE ADDRESS OF THE CSR OF THE DUT.
022366  010124                  MOV     R1,(R4)+        ;SELECT A BANK OF DUT REGISTERS.
022370  010324          6$:     MOV     R3,(R4)+        ;WRITE THE CODE TO A DUT REGISTER.
022372  020467  157664          CMP     R4,TXBFCA       ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
022376  103774                  BLO     6$              ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
022400  032701  000017          BIT     #17,R1          ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
022404  001365                  BNE     4$              ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.

022406          60$:    PASS                            ;RESTORE GPRS.
022406  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
022410  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - SWAPO -
                                ;** ************************************************************************
                                ;*                          - SWAP GPRS WITH GPR SET 0 ROUTINE -
                                ;*      THIS SUBROUTINE SWAPS THE PRESENT CONTENTS OF GPRS R1 THRU R5 WITH
                                ;*      THE CONTENTS OF THE NUMBER ZERO GPR SAVE AREA.  THE CONTENTS OF R0
                                ;*      ARE NOT ALTERED BY THIS SUBROUTINE.
                                ;*
                                ;* INPUTS:      GPR CONTENTS R1 THRU R5.
                                ;*              GPRS0B - LABEL AT BASE OF GPR SAVE AREA NUMBER ZERO.
                                ;*
                                ;* OUTPUTS:     R1 THRU R5 CONTAIN THE PREVIOUS CONTENTS OF GPR SAVE AREA
                                ;*                      ZERO WORDS 1 THRU 5 RESPECTIVELY.
                                ;*              GPRS0 - GPR SAVE AREA 0 WORDS 1 THRU 5, CONTAIN PREVIOUS
                                ;*                      CONTENTS OF GPRS R1 THRU R5 RESPECTIVELY.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,SWAPO
                                ;*
                                ;* COMMENTS:    THE STATE OF THE CARRY FLAG IS NOT ALTERRED BY THIS ROUTINE.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

022412  010046                  SWAPO:: MOV     R0,-(SP)        ;SAVE THE CONTENTS OF R0.
                                ;*
                                ; LOAD THE STACK FROM THE GPRS.
                                ;-
022414  010146                          MOV     R1,-(SP)        ;SAVE THE CONTENTS OF R1.
022416  010246                          MOV     R2,-(SP)        ;SAVE THE CONTENTS OF R2.
022420  010346                          MOV     R3,-(SP)        ;SAVE THE CONTENTS OF R3.
022422  010446                          MOV     R4,-(SP)        ;SAVE THE CONTENTS OF R4.
022424  010546                          MOV     R5,-(SP)        ;SAVE THE CONTENTS OF R5.
                                ;*
                                ; LOAD THE GPRS FROM THE GPR SAVE AREA 0.
                                ;-
022426  012700  002450                  MOV     #GPRS0B,R0      ;GET THE BASE ADDRESS OF GPR SAVE AREA 0.
022432  012001                          MOV     (R0)+,R1        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 1.
022434  012002                          MOV     (R0)+,R2        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 2.
022436  012003                          MOV     (R0)+,R3        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 3.
022440  012004                          MOV     (R0)+,R4        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 4.
022442  012005                          MOV     (R0)+,R5        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 5.
                                ;*
                                ; LOAD THE GPR SAVE AREA 0 FROM THE STACK.
                                ;-
022444  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 5 WITH SAVED R5.
022446  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 4 WITH SAVED R4.
022450  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 3 WITH SAVED R3.
022452  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 2 WITH SAVED R2.
022454  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 1 WITH SAVED R1.

022456  012600                          MOV     (SP)+,R0        ;RESTORE THE INITIAL VALUE OF R0.

022460  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.
                                ;-- ************************************************************************

022462                          TSABRT:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
022462  004567  161310                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022466  012701  022504                 MOV     #2$,R1          ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
022472  012767  017006  161274         MOV     #ER1603,ERRBLK  ;SET-UP THE ERROR REPORTING ROUTINE.
022500                          ERROR                    ;                >>>>> ERROR <<<<<.
022500  104460                                                                    TRAP    C$ERROR
022502  000432                 BR      60$             ;
022504    040     116     117  2$:     .ASCIZ / NON-RELATED TEST ERROR FOUND DURING TEST EXECUTION/
022507    116     055     122
022512    105     114     101
022515    124     105     104
022520    040     124     105
022523    123     124     040
022526    105     122     122
022531    117     122     040
022534    106     117     125
022537    116     104     040
022542    104     125     122
022545    111     116     107
022550    040     124     105
022553    123     124     040
022556    105     130     105
022561    103     125     124
022564    111     117     116
022567    000

                                .EVEN
022570                          60$:    PASS                     ;RESTORE GPRS.
022570  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
022572  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXDSBL -
                        ;++ ************************************************************************
                        ;*                          - TRANSMITTER DISABLE -
                        ;*      THIS SUBROUTINE IS USED TO DISABLE TRANSMISSION ON SELECTED LINES BY,
                        ;*      CLEARING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
                        ;*
                        ;* INPUTS:      R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO CLEAR TX.ENABLE.
                        ;*              CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
                        ;*              IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
                        ;*              NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES 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.
                        ;-- ************************************************************************

022574                  TXDSBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
022574  004567  161176                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
022600  010500                  MOV     R5,R0                   ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
022602  012701  000001          MOV     #BIT0,R1                ;INITIALIZE THE SELECTED LINE BIT MASK.
022606  016702  157446          MOV     TXAD2A,R2               ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
022612  005202                  INC     R2                      ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
022614  012703  000010          MOV     #NUMLNS,R3              ;GET MAXIMUM LINE NUMBER PLUS ONE.
022620  016704  157504          MOV     IESTAT,R4               ;GET THE STATES OF THE INT ENABLE BITS.
022624  005005                  CLR     R5                      ;LOG POSSIBLE TX DISABLED ON ALL LINES.
                        ;*
                        ; SELECT EVERY LINE IN TURN, AND LOG THE STATE OF EACH TX.ENABLE BIT.
                        ;-
022626  010477  157412  2$:      MOV     R4,@CSRA                ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
022632  105712                  TSTB    (R2)                    ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
022634  100001                 BPL     4$                      ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
022636  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.
                        ;-
022640  030100          4$:      BIT     R1,R0                   ;CHECK STATE OF DISABLE LINE BIT MAP.
022642  001402                 BEQ     6$                      ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
022644  142712  000200          BICB    #BIT7,(R2)              ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
022650  005204          6$:      INC     R4                      ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
022652  006301                  ASL     R1                      ;SHIFT BIT MAP FOR NEXT LINE.
022654  005303                  DEC     R3                      ;DECREMENT LINE NUMBER.
022656  001363                 BNE     2$                      ;LOOP TO CHECK NEXT LINE.

022660                  60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
022660  010566  000014                          MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
022664  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
                                                                ;R5 - PREVIOUS STATES OF ALL TX.ENABLE BITS.
022666  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.
                        ;-- ************************************************************************

022670                  TXENBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
022670  004567  161102                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022674  010500                  MOV     R5,R0                   ;COPY BIT MAP OF LINES TO ENABLE.
022676  012701  000001          MOV     #BIT0,R1                ;INITIALIZE THE SELECTED LINE BIT MASK.
022702  016702  157352          MOV     TXAD2A,R2               ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
022706  005202                  INC     R2                      ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
022710  012703  000010          MOV     #NUMLNS,R3              ;GET MAXIMUM LINE NUMBER.
022714  016704  157410          MOV     IESTAT,R4               ;GET THE STATES OF THE INT ENABLE BITS.
022720  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.
                        ;-
022722  010477  157316  2$:     MOV     R4,@CSRA                ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
022726  105712                  TSTB    (R2)                    ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
022730  100401                  BMI     4$                      ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
022732  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.
                        ;-
022734  030100          4$:     BIT     R1,R0                   ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
022736  001402                  BEQ     6$                      ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
022740  152712  000200          BISB    #BIT7,(R2)              ;ENABLE TRANSMISSION ON SELECTED LINE.
022744  005204          6$:     INC     R4                      ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
022746  006301                  ASL     R1                      ;SHIFT BIT MAP FOR NEXT LINE.
022750  005303                  DEC     R3                      ;DECREMENT LINE NUMBER.
022752  001363                  BNE     2$                      ;LOOP TO CHECK NEXT LINE.

022754                  60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
022754  010566  000014                          MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
022760  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;R5 - LINE BIT MAP CORRESPONDING TO THE
                                                                ; PREVIOUS LINES THAT WERE DISABLED.
022762  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - TXIEO -
                                ;++ ************************************************************************
                                ;*                          - 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,TXIEO
                                ;*
                                ;* COMMENTS:    THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*              THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************
022764  010046                  TXIEO::  MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
022766                                   GETPRI  -(SP)           ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
022766  104440                                                                   TRAP    C$GPRI
022770  010046                                                                   MOV     R0,-(SP)
022772                                   SETPRI  #PRI07          ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
022772  012700  000340                                                          MOV     #PRI07,R0
022776  104441                                                                   TRAP    C$SPRI
023000  042767  177677  157322          BIC     #177677,IESTAT  ;CLEAR TX.INT.ENBL BIT IN IESTAT.
023006  016777  157316  157230          MOV     IESTAT,@CSRA    ;DISABLE TX INTERRUPTS.
023014                                   SETPRI  (SP)+           ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
023014  012600                                                                   MOV     (SP)+,R0
023016  104441                                                                   TRAP    C$SPRI
023020  012600                          MOV     (SP)+,R0        ;RESTORE R0.
023022  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - TXIE1 -
                                ;++ ************************************************************************
                                ;*                        - TRANSMITTER INTERRUPT ENABLE -
                                ;*      THIS ROUTINE IS USED TO ENABLE TRANSMITTER INTERRUPTS IN THE DHV11.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      THE TX.INT.ENBL BIT IS SET IN THE DUT CSR.
                                ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
                                ;*               ENABLE BITS.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TXIE1
                                ;*
                                ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*               THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

023024  052767  040000  157276  TXIE1::  BIS     #BIT14,IESTAT   ;SET TX.INT.ENBL BIT IN IESTAT.
023032  042767  137677  157270           BIC     #137677,IESTAT  ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
023040  016777  157264  157176           MOV     IESTAT,@CSRA    ;DISABLE TX INTERRUPTS.
023046  000207                           RTS     PC
```

GLOBAL SUBROUTINE - UNSDIV -

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

023050                  UNSDIV:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023050  004567  160722                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
                        ;-
023054  010204                  MOV     R2,R4           ;GET MSW OF DIVIDEND FOR SUBTRACT.
023056  160104                  SUB     R1,R4           ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
023060  103403                  BCS     2$              ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
023062  012701  177777          MOV     #-1,R1          ;SET QUOTIENT TO ALL ONES (177777),
023066  000442                  BR      60$             ;EXIT WITH CARRY CLEAR.
                        ;*
                        ; SET UP COUNTERS AND VARIOUS WORKING GPRS.
                        ;-
023070  005004          2$:     CLR     R4              ;CLEAR THE LSW OF THE DIVISOR.
023072  000241                  CLC                     ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
023074  006001                  ROR     R1              ; DIVISOR BY
023076  006004                  ROR     R4              ; 2(UNSIGNED)
023100  012700  000020          MOV     #16.,R0         ;SET UP INITIAL SHIFT COUNT TO 16.
                        ;*
                        ; THE SUBTRACT AND SHIFT LOOP.
                        ;-
023104  010246          4$:     MOV     R2,-(SP)        ;SAVE MSWORD OF DIVIDEND.
023106  010346                  MOV     R3,-(SP)        ;SAVE LSWORD OF DIVIDEND.
023110  160403                  SUB     R4,R3           ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
023112  005602                  SBC     R2              ;MSWORD DIVIDEND - BORROW .
023114  103402                  BCS     6$              ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
023116  160102                  SUB     R1,R2           ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
023120  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.
                            ;-
023122  012603          6$:     MOV     (SP)+,R3        ;RESTORE LSWORD OF DIVIDEND.
023124  012602                  MOV     (SP)+,R2        ;RESTORE MSWORD OF DIVIDEND.
```

GLOBAL SUBROUTINE - UNSDIV -

```
3853 023126  000401                    BR      10$             ;GOTO SHIFT 1 INTO THE QUOTIENT.
3854                                   ;+
3855                                   ; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
3856                                   ; COMPLEMENTED LATER).   CARRY IS CLEAR.
3857                                   ;-
3858 023130  012626                8$: MOV     (SP)+,(SP)+     ;POP THE SAVED DIVIDEND OFF OF THE STACK.
3859                                   ;+
3860                                   ; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.
3861                                   ;-
3862 023132  006105                10$: ROL     R5              ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
3863 023134  000241                     CLC                     ;DIVIDE THE
3864 023136  006001                     ROR     R1              ;  DEVISOR BY
3865 023140  006004                     ROR     R4              ;  2 (UNSIGNED).
3866 023142  005300                     DEC     R0              ;COUNT THIS SHIFT AND SUBTRACT.
3867 023144  001357                     BNE     4$              ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
3868 023146  005105                     COM     R5              ;GET QUOTIENT FROM INVERTED QUOTIENT.
3869                                   ;+
3870                                   ; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.
3871                                   ;-
3872 023150  000241                     CLC                     ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
3873 023152  006103                     ROL     R3              ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
3874 023154  103402                     BCS     12$             ;IF CARRY FROM SHIFT, ROUND UP.
3875 023156  160403                     SUB     R4,R3           ;SUBTRACT DIVISOR FROM DIVIDEND.
3876 023160  103403                     BCS     14$             ;IF BORROW, DON'T ROUND UP.
3877                                   ;+
3878                                   ; ROUND UP, EXTRA SUBTRACT WENT.
3879                                   ;-
3880 023162  005205                12$: INC     R5              ;INCREMENT THE QUOTIENT BY ONE.
3881 023164  001001                     BNE     14$             ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
3882 023166  005305                     DEC     R5              ;DON'T LET ROUNDING CAUSE OVERFLOW.
3883                                   ;+
3884                                   ; ALL DONE, PASS QUOTIENT AND EXIT.
3885                                   ;-
3886 023170  010501                14$: MOV     R5,R1           ;PASS QUOTIENT BACK IN R1.
3887 023172  000261                     SEC                     ;INDICATE NO OVERFLOW.
3888
3889 023174                        60$: PASS    R1              ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
     023174  010166  000004                      MOV     R1,R1SLOT(SP)       ;PUT R1 IN STACK SLOT.
     023200  004736                      JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
3890                                                             ;R1 - 16 BIT, UNSIGNED QUOTIENT,
3891 023202  000207                     RTS     PC              ;CARRY - SLT INDICATES NO OVERFLOW (SUCCESS).
```

GLOBAL SUBROUTINE - WAIBIC -

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

023204                  WAIBIC:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023204  004567  160566                 JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
023210  010204                 MOV     R2,R4                   ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
023212  010102                 MOV     R1,R2
023214  042701  170000         BIC     #170000,R1      ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
023220  042702  007777         BIC     #7777,R2        ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
023224  000302                 SWAB    R2              ;PUT LINE NUMBER FIELD IN LSBYTE.
023226  006202                 ASR     R2              ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
023230  006202                 ASR     R2              ;  POSITION TO USE IT AS A WORD TABLE OFFSET
023232  006202                 ASR     R2              ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
023234  016202  002410         MOV     BITTBL(R2),R2   ;GET BIT MAP OF LINE TO TEST FROM TABLE.
023240  005003                 CLR     R3              ;INDICATE THAT THE BIT SHOULD BE CLR.
023242  004767  175070         JSR     PC,MSLGET       ;WAIT FOR THE BIT TO BE CLR WITHIN TIME-OUT.
                                                        ;  CARRY IS CORRECT UPON MSLGET RETURN.
023246  010002                 MOV     R0,R2           ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
023250                  60$:    PASS    R2              ;RESTORE GPRS, EXCEPT THE FOLLOWING:
023250  010266  000006                         MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
023254  004736                         JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
                                                        ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
023256  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.
                                ;-- ************************************************************************

023260                          WAIBIS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023260  004567  160512                   JSR     R5,PREGO5               ;CALL REGISTER SAVE SUBRT.
023264  010204                           MOV     R2,R4                   ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
023266  010102                           MOV     R1,R2
023270  042701  170000                   BIC     #170000,R1              ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
023274  042702  007777                   BIC     #7777,R2                ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
023300  000302                           SWAB    R2                      ;PUT LINE NUMBER FIELD IN LSBYTE.
023302  006202                           ASR     R2                      ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
023304  006202                           ASR     R2                      ;  POSITION TO USE IT AS A WORD TABLE OFFSET
023306  006202                           ASR     R2                      ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
023310  016202  002410                   MOV     BITTBL(R2),R2           ;GET BIT MAP OF LINE TO TEST FROM TABLE.
023314  010203                           MOV     R2,R3                   ;INDICATE THAT THE BIT SHOULD BE SET.
023316  004767  175014                   JSR     PC,MSLGET               ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
                                                                         ;  CARRY IS CORRECT UPON MSLGET RETURN.
023322  010002                           MOV     R0,R2                   ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
023324                          60$:     PASS    R2                      ;RESTORE GPRS, EXCEPT THE FOLLOWING:
023324  010266  000006                           MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
023330  004736                           JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
                                                                         ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
023332  000207                           RTS     PC                      ;  CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).
```

```
                        .SBTTL  GLOBAL SUBROUTINE                       - WDPDR -
                        ;** ************************************************************************
                        ;*                       - WRITE DATA PATTERN TO DEVICE REGISTERS -
                        ;*      THIS ROUTINE WRITES A ROTATED DATA PATTERN TO EACH OF THE 6 DEVICE
                        ;*      REGISTERS OF EACH ACTIVE LINE OF THE DEVICE UNDER TEST.
                        ;*      THE DATA PATTERN IS ROTATED ONCE AFTER EACH WRITE TO A DEVICE REGISTER
                        ;*              ON A PARTICULAR LINE.  THE STARTING DATA PATTERN FOR EACH LINE
                        ;*              IS ROTATED ONCE AFTER WRITING ALL THE REGISTERS ON A PARTICULAR
                        ;*              LINE.  THIS LEADS TO THE FOLLOWING DATA PATTERN:
                        ;*                      LINE 0, REGISTER 0 - SHIFTED 0 BIT POSITIONS
                        ;*                      LINE 0, REGISTER 1 - SHIFTED 1 BIT POSITION
                        ;*                      ...
                        ;*                      LINE 1, REGISTER 0 - SHIFTED 1 BIT POSITION
                        ;*                      LINE 2, REGISTER 1 - SHIFTED 2 BIT POSITIONS
                        ;*                      ...
                        ;*      ANY BITS FIELDS IN THE DEVICE REGISTERS THAT CANNOT BE ALTERED
                        ;*      ARE MASKED OUT OF THE DATA PATTERN BEFORE IT IS WRITTEN.
                        ;*      THIS ROUTINE WILL USE EITHER MOV, MOVB, BIS, BISB, BIC, OR BICB
                        ;*      INSTRUCTIONS.  THE UPPER OR LOWER BYTE CAN BE SPECIFIED FOR WRITING.
                        ;*
                        ;* INPUTS:      R2 - USED TO PASS IN THE DATA PATTERN TO BE ROTATED & WRITTEN.
                        ;*              R3 - BYTE INDICATOR (- => LO BYTE, + => HI BYTE, 0 => BOTH).
                        ;*              R4 - OPERATION TYPE INDICATOR (- => BIC, + => BIS, 0 => MOV).
                        ;*              ACTLNS - BIT MAP OF THE ACTIVE LINES ON THE DEVICE UNDER TEST.
                        ;*              CSRA - CONTAINS THE CSR ADDRESS OF THE DEVICE UNDER TEST.
                        ;*              DRADRT - BASE ADDRESS OF DEVICE REGISTER ADDRESS TABLE.
                        ;*              LPRO - EQUATED TO LPR REG OFFSET FROM DEVICE CSR ADDRESS.
                        ;*              NUMLNS - NUMBER OF LINES ON THE DEVICE UNDER TEST.
                        ;*              TXBFCO - EQUATED TO TBUFFCT REG OFFSET FROM DEVICE CSR ADDRESS.
                        ;*              UNBTTB - BASE ADDRESS OF THE UNUSED BIT TABLE.
                        ;*
                        ;* OUTPUTS:     DEVICE REGISTERS ON ALL ACTIVE DEVICE LINES ARE MODIFIED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WDPDR
                        ;*
                        ;* COMMENTS:    THIS ROUTINE DOES NOT WRITE ANY DATA TO THE TX.CHAR REGISTERS.
                        ;*              THE CSR IS CLEARED EXCEPT FOR THE IND.ADR.REG FIELD.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ROLDAP.
                        ;-- ************************************************************************

023334                  WDPDR:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
023334  004567  160436                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; SET UP OUTER LOOP WHICH WRITES THE DATA PATTERN TO EACH LINE'S REGISTERS
                        ;-
023340  005005                  CLR     R5                      ;CLEAR LINE COUNTER TO SELECT LINE 0.
                        ;+
                        ; THE OUTER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP WRITES DATA TO ALL OF
                        ; THE DEVICE REGISTERS FOR A PARTICULAR LINE IF IT IS ACTIVE.
                        ;-
023342  010204          2$:     MOV     R2,R4                   ;SAVE THE OUTER LOOP DATA PATTERN.
023344  010577  156674          MOV     R5,@CSRA                ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
023350  006305                  ASL     R5                      ;TURN LINE NUMBER INTO A WORD OFFSET.
023352  036567  002410  156656          BIT     BITTBL(R5),ACTLNS
023360  001451                  BEQ     20$                     ;LINE ACTIVE?  NO, SKIP THIS LINE.
023362  012701  000004          MOV     @LPRO,R1                ;YES, INITIALIZE THE REGISTER OFFSET.
```

```
                                        ;*
                                        ;       THE INNER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP WRITES DATA TO A
                                        ;       DEVICE REGISTER.
                                        ;-
023366  010200                  4$:       MOV     R2,R0
023370  046100  002264                    BIC     UNBITB(R1),R0   ;CLEAR BIT FIELDS FOR UNUSED REGISTER BITS.
023374  016103  002244                    MOV     DRADRT(R1),R3   ;GET THE ADDRESS OF THE DEVICE REGISTER.
023400  005766  000010                    TST     R3SLOT(SP)      ;CHECK THE OPERAND TYPE INDICATOR.
023404  003402                          BLE     6$              ;HIGH BYTE?  NO, SKIP HIGH BYTE ADDRESS SET UP.
023406  005203                           INC     R3              ;YES, SET THE REG ADDRESS TO THE HIGH BYTE.
023410  000300                           SWAB    R0              ;MOVE HIGH BYTE DATA INTO THE LOW BYTE.
023412  005766  000010          6$:      TST     R3SLOT(SP)      ;CHECK THE OPERAND TYPE INDICATOR.
023416  001412                          BEQ     12$             ;WORD ACCESS?  YES, GO PERFORM WORD ACCESS.
                                    ;*
                                    ;PERFORM BYTE ACCESS TO THE SPECIFIED BYTE OF THE SPECIFIED REGISTER.
                                    ;-
023420  005766  000012                    TST     R4SLOT(SP)      ;NO, CHECK THE ACCESS TYPE INDICATOR.
023424  100403                          BMI     8$              ;USE BIC?  YES, GO PERFORM BICB INSTRUCTION.
023426  001404                          BEQ     10$             ;USE MOV?  YES, GO PERFORM MOVB INSTRUCTION.
023430  150013                           BISB    R0,(R3)         ;NEITHER.  PERFORM BISB ACCESS TO REGISTER.
023432  000415                          BR      18$
023434  140013                  8$:      BICB    R0,(R3)         ;PERFORM BICB ACCESS TO REGISTER.
023436  000413                          BR      18$
023440  110013                  10$:     MOVB    R0,(R3)         ;PERFORM MOVB ACCESS TO REGISTER.
023442  000411                          BR      18$
                                    ;*
                                    ;PERFORM WORD ACCESS TO THE SPECIFIED REGISTER.
                                    ;-
023444  005766  000012          12$:      TST     R4SLOT(SP)      ;CHECK THE ACCESS TYPE INDICATOR.
023450  100403                          BMI     14$             ;USE BIC?  YES, GO PERFORM BIC INSTRUCTION.
023452  001404                          BEQ     16$             ;USE MOV?  YES, GO PERFORM MOV INSTRUCTION.
023454  050013                           BIS     R0,(R3)         ;NEITHER.  PERFORM BIS ACCESS TO REGISTER.
023456  000403                          BR      18$
023460  040013                  14$:     BIC     R0,(R3)         ;PERFORM BIC ACCESS TO REGISTER.
023462  000401                          BR      18$
023464  010013                  16$:     MOV     R0,(R3)         ;PERFORM MOV ACCESS TO REGISTER.
                                    ;*
                                    ; PREPARE THE DATA PATTERN AND OFFSET FOR THE NEXT REGISTER ON THIS LINE.
                                    ;-
023466  004767  176074          18$:     JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
023472  062701  000002                    ADD     #2,R1           ;INCREMENT OFFSET FOR NEXT REGISTER.
023476  020127  000016                    CMP     R1,#TXBFCO      ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
023502  003731                          BLE     4$              ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
                                    ;*
                                    ; BACK INTO THE OUTER LOOP.  NOW SET UP FOR NEXT LINE.  LOOP IF NOT DONE.
                                    ;-
023504  010402                  20$:     MOV     R4,R2           ;SET UP TO ROTATE THE DATA PATTERN.
023506  004767  176054                    JSR     PC,ROLDAP       ;ROTATE THE DATA PATTERN.
023512  006205                           ASR     R5              ;CONVERT BACK TO LINE NUMBER FROM WORD OFFSET.
023514  005205                           INC     R5              ;COUNT THIS LINE.
023516  020527  000010                    CMP     R5,#NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
023522  002707                           BLT     2$              ;LOOP IF SOME LINES NOT DONE.

023524                          60$:     PASS                    ;RESTORE GPRS.
023524  004736                                           JSR     PC,@(SP)+               ;RETURN TO PREGCS SUBRT.
023526  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.
                        ;-- ************************************************************************

023530                  WTWLNC:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023530  004567  160242                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
                        ;-
023534  016701  156514          MOV     LNCTRA,R1       ;SET UP THE REGISTER ADDRESS PARAMETER.
023540  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
023542  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
023544  012704  177777          MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                        ;*
                        ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
                        ;-
023550  004767  174052          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

023554                  60$:    PASS                    ;RESTORE GPRS.
023554  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023556  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - WTWLNS -
                        ;*****************************************************************************
                        ;*                      - WRITE WORD TO ALL LINES ROUTINE -
                        ;*      THIS SUBROUTINE WRITES A SPECIFIED WORD TO THE SPECIFIED DHV DEVICE
                        ;*      REGISTER FOR EACH OF THE DHV LINES.  IT COULD BE USED TO CLEAR ALL
                        ;*      OF THE LNCTRL REGISTERS OR TO INITIALIZE ALL OF THE LPR REGISTERS TO
                        ;*      THE SAME PARAMETERS.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF THE SPECIFIED REGISTERS.
                        ;*              R2 - WORD TO WRITE INTO THE SPECIFIED REGISTERS.
                        ;*              IESTAT - SAVED STATES OF THE TX.IE AND RX.IE BITS.
                        ;*              MAPLNS - EQUATED TO BIT MAP OF LINES ON DEVICE (8 FOR DHV11).
                        ;*                CSRA.
                        ;*
                        ;* OUTPUTS:     DEVICE REGISTERS - SPECIFIED REGISTERS GIVEN NEW VALUE.
                        ;*              CSR IND.ADR.REG FIELD - DESTROYED.
                        ;*              CSR INTERRUPT ENABLE BITS - SET TO STATES IN IESTAT.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WTWLNS
                        ;*
                        ;* COMMENTS:    NOTE THAT THE SPECIFIED REGISTERS FOR ALL LINES ARE ALTERED
                        ;*              BY THIS ROUTINE.  THIS ROUTINE SHOULD NOT BE USED TO ALTER
                        ;*              THE STATES OF PARTIAL REGISTER FIELDS OR TO ALTER A REGISTER
                        ;*              FOR FEWER THAN ALL OF THE LINES.
                        ;*              THE SPECIFIED REGISTERS ARE READ BEFORE BEING WRITTEN.
                        ;*
                        ;* SUBROUTINES CALLED: ALTFLD.
                        ;*****************************************************************************

023560                  WTWLNS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5
023560  004567  160212                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; SET UP THE BIT MAP OF LINES TO CHANGE AND MASK OF BITS TO ALTER PARAMETERS.
                        ;-
023564  012703  000377          MOV     @MAPLNS,R3      ;GET THE BIT MAP OF LINES TO CHANGE.
023570  012704  177777          MOV     @-1,R4          ;INDICATE ALL 16 BITS TO BE CHANGED.
                        ;*
                        ; CALL THE SUBROUTINE TO WRITE THE SPECIFIED REGISTERS.
                        ;-
023574  004767  174026          JSR     PC,ALTFLD       ;CHANGE THE REGISTERS.

023600                  60$:    PASS                            ;RESTORE GPRS.
023600  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023602  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.
                        ;-- ************************************************************************

023604                  WTWLPR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023604  004567  160166                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
                        ;-
023610  016701  156434          MOV     LPRA,R1         ;SET UP THE REGISTER ADDRESS PARAMETER.
023614  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
023616  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
023620  012704  177777          MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                        ;*
                        ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
                        ;-
023624  004767  173776          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

023630                  60$:    PASS                    ;RESTORE GPRS.
023630  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023632  000207          RTS     PC
```

```
                        .SBTTL  INTERRUPT SERVICE ROUTINE       - CACHRX -
                        ;++ ************************************************************************
                        ;*      - CATCH RECEIVER INTERRUPT.
                        ;*      THIS ROUTINE IS USED IN SEVERAL TESTS,TO LOG A COUNT OF THE
                        ;*      NUMBER OF RECEIVER INTERUPTS THAT OCCUR.
                        ;*
                        ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR.
                        ;*              RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERRUPTS
                        ;*                       THAT OCCURRED.
                        ;*
                        ;* OUTPUTS:     RXINTC - CONTAINS THE UPDATED INTERRUPT COUNT.
                        ;*
                        ;*
                        ;* CALLING SEQUENCE:    PUT THE ADDRESS OF THE LABEL CACHRX IN THE VECTOR
                        ;*                      LOCATION.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE
                        ;-- ************************************************************************

023634                  CACHRX::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
023634  004567  160136          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023640  016701  156470          MOV     RXINTC,R1       ;GET THE RECEIVER INTERRUPT COUNT
023644  005201                  INC     R1              ;INCREMENT THE COUNT
023646  102001                  BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED
023650  005301                  DEC     R1              ;RESET THE COUNT TO 177777
023652  010167  156456  2$:     MOV     R1,RXINTC       ;SAVE NEW COUNT VALUE
023656                  60$:    PASS                    ;RESTORE GPRS.
023656  004736                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023660  000002                  RTI
```

```
                        .SBTTL  INTERRUPT SERVICE ROUTINE       - CACHTX -
                        ;** ************************************************************************
                        ;*      - CATCH TRANSMITER INTERRUPT.
                        ;*      THIS ROUTINE IS USED IN SEVERAL TESTS,TO LOG A COUNT OF THE
                        ;*      NUMBER OF TRANSMISSION INTERRUPTS THAT OCCUR.
                        ;*
                        ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR.
                        ;*              TXINTC - HOLDS THE COUNT OF THE NUMBER OF TX INTERRUPTS
                        ;*                       THAT OCCURRED.
                        ;*
                        ;* OUTPUTS:     TXINTC - CONTAINS THE UPDATED INTERRUPT COUNT.
                        ;*
                        ;*
                        ;* CALLING SEQUENCE:    PUT THE ADDRESS OF THE LABEL CACHTX IN THE VECTOR
                        ;*                      LOCATION.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE
                        ;-- ************************************************************************

023662                  CACHTX::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
023662  004567  160110          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023666  016701  156446          MOV     TXINTC,R1       ;GET THE TRANSMISSION INTERRUPT COUNT
023672  005201                  INC     R1              ;INCREMENT THE COUNT
023674  102001                  BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED
023676  005301                  DEC     R1              ;RESET THE COUNT TO 177777
023700  010167  156434  2$:     MOV     R1,TXINTC       ;SAVE NEW COUNT VALUE
023704                  60$:    PASS                    ;RESTORE GPRS.
023704  004736                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023706  000002                  RTI
```

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

023710  005767  156450        CLKINT:: TST     TIMER1          ;CHECK FOR TIMER1 AT ZERO.
023714  001402                         BEQ     2$              ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
023716  005367  156442                 DEC     TIMER1          ;DECREMENT TIME COUNT.
023722  005767  156440        2$:      TST     TIMER2          ;CHECK FOR TIMER2 AT ZERO.
023726  001402                         BEQ     4$              ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
023730  005367  156432                 DEC     TIMER2          ;DECREMENT TIME COUNT.
023734  005367  156430        4$:      DEC     TIMER3          ;DECREMENT THE BREAK COUNT.
023740  001006                         BNE     60$             ;EXIT IF NOT TIME TO CALL BREAK.
023742  016767  156424  156420        MOV     BCOUNT,TIMER3   ;SET UP TIME TILL NEXT BREAK.
023750  010046                         MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 FROM BREAK MACRO.
023752                                 BREAK                   ;CHECK FOR OPERATOR CONTROL/C.
023752  104422                                                         TRAP    C$BRK
023754  012600                         MOV     (SP)+,R0        ;RESTORE CONTENTS OF R0.
023756  000002                60$:     RTI
```

INTERUPT SERVICE ROUTINE - RXBRRT -

```
                        .SBTTL  INTERUPT SERVICE ROUTINE        - RXBRRT -
                        ;** ************************************************************************
                        ;*      - BR LEVEL TEST RECEIVE INTERRUPT SERVICE ROUTINE -
                        ;*      THIS SERVICE ROUTINE HANDLES RECEIVE INTERRUPTS DURING THE INTERRUPT
                        ;*      BR LEVEL TEST.  THIS ROUTINE COUNTS THE INTERRUPT AND SETS A FLAG
                        ;*      TO INDICATE THAT THE INTERRUPT HAS OCCURRED.  IT ALSO CHECKS THE
                        ;*      FLAG WHICH INDICATES THAT A TX INTERRUPT HAS OCCURRED.  IF THE TX
                        ;*      INTERRUPT FLAG IS SET, THIS ROUTINE SETS AN INTERRUPT ORDER ERROR
                        ;*      FLAG INDICATING THAT A TRANSMIT INTERRUPT WAS SERVICED BEFORE A
                        ;*      SIMULTANEOUS RECEIVE INTERRUPT.
                        ;*
                        ;* INPUTS:      RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERUPTS.
                        ;*              RXINTF - RX INTERRUPT FLAGS.
                        ;*
                        ;* OUTPUTS:     RXINTC - CONTAINS THE UPDATED INTERUPT COUNT.
                        ;*              RXINTF - RX INT FLAGS:
                        ;*                       (BIT 0 SET, BIT 14 SET IF TXINTF BIT 0 IS SET.)
                        ;*
                        ;* CALLING SEQUENCE:    PUT THE ADDRESS OF THE LABEL RXBRRT IN THE VECTOR
                        ;*                      LOCATION.
                        ;*
                        ;* COMMENTS:    NOTE: THE FIFO IS PURGED BY THIS ROUTINE.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

023760                  RXBRRT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023760  004567  160012                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023764  017700  156256          MOV     @RBUFA,R0               ;READ THE CHAR OUT OF THE FIFO.
023770  016701  156340          MOV     RXINTC,R1               ;GET THE INTERUPT COUNT.
023774  005201                  INC     R1                      ;INCREMENT THE COUNT.
023776  001402                  BEQ     2$                      ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
024000  010167  156330          MOV     R1,RXINTC               ;SAVE NEW COUNT VALUE.
024004  016701  156326  2$:     MOV     RXINTF,R1               ;GET THE RX INTERRUPT FLAGS.
024010  052701  000001          BIS     #BIT0,R1                ;SET THE RX INTERRUPT HAS OCCURRED FLAG.
024014  032767  000001  156320  BIT     #BIT0,TXINTF            ;TEST THE "TX INT HAS OCCURRED" FLAG.
024022  001402                  BEQ     4$                      ;SKIP SETTING ERROR FLAG IF NO TX INT.
024024  052701  040000          BIS     #BIT14,R1               ;SET THE INTERRUPT ORDER ERROR FLAG.
                        ;+
                        ; 8 FIFO CODES WILL CAUSE 8 INTERRUPTS, AFTER THESE 8 CODES WE DON'T WANT
                        ;  TO CHECK THE INTERRUPT ORDER, BECAUSE PERHAPS A BMP CODE HAS COME IN
                        ;  BETWEEN THE SERVICING OF THE 8 FIFO CODE INTERRUPTS AND THE SERVICING
                        ;  OF ONE OF THE TX INTERRUPTS.
                        ;-
024030  026767  156300  153752  4$:     CMP     RXINTC,NUMLNS   ;TEST FOR ALL SELFTEST CODE INTS DONE.
024036  003002                  BGT     60$                     ;SKIP UPDATING RX INT FLAGS IF EXTRA RX INTS.
024040  010167  156272          MOV     R1,RXINTF               ;UPDATE THE RX INTERRUPT FLAGS.
024044                  60$:    PASS                            ;RESTORE GPRS.
024044  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
024046  000002                  RTI
```

INTERUPT SERVICE ROUTINE - RXINPT -

```
                        .SBTTL  INTERUPT SERVICE ROUTINE        - RXINPT -
                        ;** ************************************************************************
                        ;*      - RECEIVE CHARACTER INPUT INTERRUPT SERVICE ROUTINE -
                        ;*      THIS SERVICE ROUTINE INPUTS A CHARACTER FROM THE DUT AND LOADS THE
                        ;*      CHAR (COMPLETE WITH STATUS FLAGS) INTO A RECEIVE CHAR BUFFER IN
                        ;*      MEMORY.  THE INTERRUPT IS ALSO COUNTED.  THE RECEIVE CHAR BUFFER IS
                        ;*      MONITORED TO ENSURE THAT IT DOES NOT OVERFLOW.
                        ;*
                        ;* INPUTS:      BUFEND - LABELS THE END OF THE HOST MEMORY BUFFER.
                        ;*              BUFPTR - CONTAINS ADDRESS OF NEXT FREE BUFFER LOCATION.
                        ;*              CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
                        ;*              RBUFA - CONTAINS THE ADDRESS OF THE RBUF DUT REGISTER.
                        ;*              RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERUPTS.
                        ;*              RXINTF - RX INTERRUPT FLAGS.
                        ;*
                        ;* OUTPUTS:     BUFPTR - CONTAINS UPDATED ADDRESS OF NEXT FREE BUFFER LOCATION.
                        ;*              RXINTC - CONTAINS THE UPDATED INTERUPT COUNT.
                        ;*              RXINTF - RX INT FLAGS (BIT 15 SET IF RX.DATA.AVAIL IS CLEAR).
                        ;*
                        ;* CALLING SEQUENCE:    PUT THE ADDRESS OF THE LABEL RXINPT IN THE VECTOR
                        ;*                      LOCATION.
                        ;*
                        ;* COMMENTS:    IN CASE OF OVERFLOW OF THE MEMORY BUFFER, BUFPTR WILL BE
                        ;*              MAINTAINED EQUAL TO BUFEND AND THE WORD AT BURFPTR WILL BE
                        ;*              THE LAST WORD READ FROM THE DUT FIFO.
                        ;*              NOTE: THIS ROUTINE CAN DESTROY TX.ACTIONS BY READING THE CSR.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

024050                  RXINPT:: SAVE                           ;SAVE CONTENTS OF GPRS RO THRU R5.
024050 004567 157722             JSR     R5,PREGO5              ;CALL REGISTER SAVE SUBRT.
024054 032777 000200 156162      BIT     #BIT7,@CSRA            ;TEST RX.DATA.AVAIL BIT OF THE CSR (READS CSR).
024062 001003                    BNE     2$                     ;BRANCH AROUND SETTING FLAG IF BIT IS SET.
024064 052767 100000 156244      BIS     #BIT15,RXINTF          ;SET THE RX.DATA.AVAIL CLEAR FLAG.
024072 016701 156236     2$:     MOV     RXINTC,R1              ;GET THE INTERUPT COUNT.
024076 005201                    INC     R1                     ;INCREMENT THE COUNT.
024100 001402                    BEQ     4$                     ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
024102 010167 156226             MOV     R1,RXINTC              ;SAVE NEW COUNT VALUE.
024106 016702 156212     4$:     MOV     BUFPTR,R2              ;GET THE POINTER TO NEXT FREE BUFFER WORD.
024112 017722 156130             MOV     @RBUFA,(R2)+           ;READ A CHAR FROM THE FIFO INTO BUFFER.
024116 020267 157604             CMP     R2,BUFEND              ;TEST FOR POINTER BEYOND END OF BUFFER.
024122 103002                    BHIS    60$                    ;SKIP THE PTR UPDATE IF PTR OUT OF BOUNDS.
024124 010267 156174             MOV     R2,BUFPTR              ;UPDATE THE BUFFER POINTER.
024130                  60$:     PASS                           ;RESTORE GPRS.
024130 004736                            JSR     PC,@(SP)+              ;RETURN TO PREGO5 SUBRT.
024132 000002                    RTI
```

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

024134  021627  020136        TP4RTN:: CMP     (SP),@ADRPTR    ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
024140  001402                         BEQ     2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
024142  000177  156176                 JMP     @TP4VEC         ;IF NOT,JUMP TO NORMAL 004 TRAP SERVICE RTN.
024146  052767  100000  156172  2$:    BIS     #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
024154  000002                         RTI                     ;ALL DONE, GO BACK TO THE TEST.
```

```
                        .SBTTL  INTERUPT SERVICE ROUTINE         - TXINTR -
                        ;++ ************************************************************************
                        ;*      - TRANSMIT INTERRUPT SERVICE ROUTINE -
                        ;*      THIS ROUTINE HANDLES A TRANSMIT INTERRUPT FROM THE DEVICE UNDER TEST
                        ;*      (DUT) BY COUNTING THE INTERRUPT AND READING THE DUT CSR TO CLEAR THE
                        ;*      INTERRUPT REQUEST.  THIS ROUTINE ALSO SETS A FLAG TO INDICATE THAT
                        ;*      A TX INTERRUPT HAS OCCURRED AND SETS A FLAG IF THE TX.ACTION BIT IS
                        ;*      NOT SET IN THE READ CONTENTS OF THE DUT CSR.
                        ;*
                        ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR.
                        ;*              TXINTC - HOLDS THE COUNT OF THE NUMBER OF TX INTERUPTS.
                        ;*              TXINTF - TX INTERRUPT FLAGS.
                        ;*
                        ;* OUTPUTS:     TXINTC - CONTAINS THE UPDATED TX INTERUPT COUNT.
                        ;*              TXINTF - TX INT FLAGS (BIT 0 SET, BIT 15 SET IF TX.ACTION CLR).
                        ;*
                        ;* CALLING SEQUENCE:    PUT THE ADDRESS OF THE LABEL TXINTR IN THE VECTOR
                        ;*                      LOCATION.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE
                        ;-- ************************************************************************

024156                  TXINTR:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
024156  004567  157614                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
024162  016701  156152          MOV     TXINTC,R1       ;GET THE TX INTERUPT COUNT.
024166  005201                  INC     R1              ;INCREMENT THE COUNT.
024170  102001                BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED.
024172  005301                  DEC     R1              ;RESET THE COUNT TO 177777.
024174  010167  156140  2$:     MOV     R1,TXINTC       ;SAVE NEW COUNT VALUE.
024200  016703  156136          MOV     TXINTF,R3       ;GET THE TX INTERRUPT FLAGS.
024204  017702  156034          MOV     @CSRA,R2        ;READ THE CSR.
024210  100402                BMI     4$              ;SKIP SETTING OF FLAG IF TX.ACTION IS SET.
024212  052703  100000          BIS     #BIT15,R3       ;SET THE TX.ACTION CLEAR FLAG.
024216  052703  000001  4$:     BIS     #BIT0,R3        ;SET THE TX INT HAS OCCURRED FLAG.
024222  010367  156114          MOV     R3,TXINTF       ;UPDATE THE TX INTERRUPT FLAGS.
024226                  60$:    PASS                    ;RESTORE GPRS.
024226  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
024230  000002                RTI
```

```
                        ;*****************************************************************************
                        ;
                        ;               FVTA.RPT
                        ;
                        ;*****************************************************************************


                        .SBTTL  REPORT CODING SECTION

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

024232                          BGNRPT
024232                                                          L$RPT::

024232                          EXIT    RPT
024232  000167                                                          .WORD   J$JMP
024234  000000                                                          .WORD   L10017-2-.

                                .EVEN

024236                          ENDRPT
024236                                                          L10017:
024236  104425                                                          TRAP    C$RPT
```

PROTECTION TABLE

```
4503                                    .SBTTL  PROTECTION TABLE
4504
4505                                    ;*****************************************************************************************
*
4506                                    ;
4507                                    ;               FVTSKL4.P11
4508                                    ;
4509                                    ;*****************************************************************************************
4510
4511
4512
4513                                    ;++
4514                                    ; THIS TABLE IS USED BY THE RUNTIME SERVICES
4515                                    ; TO PROTECT THE LOAD MEDIA.
4516                                    ;--
4517
4518 024240                                     BGNPROT
    024240                                                                      L$PROT::
4519
4520 024240  177777                             -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
4521 024242  177777                             -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
4522 024244  177777                             -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
4523
4524 024246                                     ENDPROT
4525
```

PROTECTION TABLE

```
4540
4541                            ;*****************************************************************************
4542                            ;
4543                            ;               FVTA.INI
4544                            ;
4545                            ;*****************************************************************************
4546
4547
4548
4549                            .SBTTL  INITIALIZE SECTION
4550                            ;++
4551                            ;*****************************************************************************
4552                            ;*      THIS SECTION CONTAINS THE CODE WHICH IS PERFORMED AT THE BEGINNING OF
4553                            ;*      EACH PASS OR AFTER A CONTINUE COMMAND.
4554                            ;*      THIS CODE PERFORMS THE FOLLOWING ACTIONS:
4555                            ;*
4556                            ;*      MOVES THE INFORMATION HELD IN THE HARDWARE P-TABLE INTO THE GLOBAL
4557                            ;*      DATA AREA.
4558                            ;*
4559                            ;*****************************************************************************
4560                            ;--
4561 024246                            BGNINIT
      024246                                                                    L$INIT::
4562
4563 024246                     ;SEE IF PROGRAM JUST STARTED, BR IF YES
                                        READEF  @EF.START
      024246  012700  000040                                                    MOV     @EF.START,R0
      024252  104447                                                            TRAP    C$REFG
4564 024254                             BCOMPLETE       NEWSTA
      024254  103416                                                            BCS     NEWSTA
4565
4566 024256                     ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
                                        READEF  @EF.RESTART
      024256  012700  000037                                                    MOV     @EF.RESTART,R0
      024262  104447                                                            TRAP    C$REFG
4567 024264                             BCOMPLETE       NEWRES
      024264  103556                                                            BCS     NEWRES
4568
4569 024266                     ;SEE IF THIS IS A NEW PASS, BR IF YES
                                        READEF  @EF.NEW
      024266  012700  000035                                                    MOV     @EF.NEW,R0
      024272  104447                                                            TRAP    C$REFG
4570 024274                             BCOMPLETE       NEWPAS
      024274  103555                                                            BCS     NEWPAS
4571
4572 024276                     ;SEE IF PROGRAM WAS JUST CONTINUED
                                        READEF  @EF.CONTINUE
      024276  012700  000036                                                    MOV     @EF.CONTINUE,R0
      024302  104447                                                            TRAP    C$REFG
4573 024304                             BNCOMPLETE      GETPRM
      024304  103161                                                            BCC     GETPRM
4574 024306  000167  000552            JMP     ENDIT
4575 024312                     NEWSTA:
4576 024312                             BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
      024312  104433                                                            TRAP    C$RESET
4577                            ;+
4578                            ; SET UP FOR LINE TIME CLOCK INTERRUPTS.
4579                            ;-
4580 024314                             CLOCK   L,R1            ;GET THE CLOCK PARAMETERS.
      024314  012700  000114                                                    MOV     @'L,R0
      024320  104462                                                            TRAP    C$CLCK
```

INITIALIZE SECTION

```
        024322  010001                                                          MOV     R0,R1
4581    024324  012167  156024                  MOV     (R1)+,CLKCSR            ;STORE CLOCK CSR ADDRESS.
4582    024330  012167  156022                  MOV     (R1)+,CLKBRL            ;STORE CLOCK BUS REQ INT LEVEL.
4583    024334  012167  156020                  MOV     (R1)+,CLKVEC            ;STORE CLOCK INTERRUPT VECTOR.
4584    024340  012167  156016                  MOV     (R1)+,CLKHRZ            ;STORE CLOCK FREQUENCY.
4585    024344  026727  156012  000062          CMP     CLKHRZ,#50.             ;TEST FOR 50HZ LINE FREQUENCY.
4586    024352  001004                          BNE     2$                      ;BRANCH IF CLOCK IS NOT 50HZ.
4587    024354  012767  000024  156012          MOV     #20.,MSTICK             ;INDICATE 20MS PER CLOCK TICK.
4588    024362  000403                          BR      4$
4589    024364  012767  000021  156002  2$:     MOV     #17.,MSTICK             ;INDICATE 17 MS PER CLOCK TICK.
4590    024372                          4$:     SETVEC  CLKVEC,#CLKINT,PRI06    ;INITIALIZE CLOCK INTERRUPT VECTOR.
        024372  016746  153702                                                          MOV     PRI06,-(SP)
        024376  012746  023710                                                          MOV     #CLKINT,-(SP)
        024402  016746  155752                                                          MOV     CLKVEC,-(SP)
        024406  012746  000003                                                          MOV     #3,-(SP)
        024412  104437                                                                  TRAP    C$SVEC
        024414  062706  000010                                                          ADD     #10,SP
4591    024420  016700  155736                  MOV     CLKHRZ,R0               ;INITIALIZE THE BREAK COUNT
4592    024424  006300                          ASL     R0                      ; TO CAUSE A BREAK
4593    024426  010067  155740                  MOV     R0,BCOUNT               ; EVERY 2 SECONDS.
4594    024432                                  SETPRI  #PRIO5                  ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
        024432  012700  000240                                                          MOV     #PRIO5,R0
        024436  104441                                                                  TRAP    C$SPRI
4595                                    ;+
4596                                    ; ENABLE THE LINE TIME CLOCK (LTC) CHECKING TO MAKE SURE THAT THE CSR
4597                                    ; IS ACCESSABLE.
4598                                    ;    FIRST SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
4599                                    ;-
4600    024440  016767  153340  155676            MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
4601    024446  012767  024134  153330            MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4602                                    ;+
4603                                    ;    ENABLE LTC CHECKING FOR 004 TRAP IN CASE CSR IS NOT THERE.
4604                                    ;-
4605    024454  005067  155666                    CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
4606    024460  012767  000100  155662            MOV     #BIT6,WORD1     ;SET UP TO SET BIT6 OF THE LTC CSR.
4607    024466  012700  002350                    MOV     #WORD1,R0       ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
4608    024472  016701  155656                    MOV     CLKCSR,R1       ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
4609    024476  004767  173422                   JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
4610    024502  016767  155636  153274            MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4611    024510  103403                            BCS     6$              ;IF NO TRAP, LTC IS THERE SO CONTINUE.
4612    024512  005067  155644                    CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
4613    024516  000402                            BR      8$              ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
4614                                    ;+
4615                                    ; CALIBRATE THE DELAY ROUTINE MILLI-SECOND DELAY COUNT VALUE.
4616                                    ;-
4617    024520  004767  173154          6$:       JSR     PC,CALMSL
4618                                    ;+
4619                                    ; CHECK FOR MEMMORY MANAGEMENT PRESENT ON THIS MACHINE.
4620                                    ; IF MEM MGT IS PRESENT, DISABLE IT.
4621                                    ;-
4622    024524  016767  153254  155612  8$:       MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
4623    024532  012767  024134  153244            MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4624    024540  005067  155602                    CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
4625    024544  005067  155600                    CLR     WORD1           ;PREPARE TO CLEAR THE MEM MGT SR0 REGISTER.
4626    024550  012700  002350                    MOV     #WORD1,R0       ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
4627    024554  016701  155620                    MOV     MMSR0,R1        ;SELECT MEM MGT SR0 REGISTER AS DESTINATION.
4628    024560  005067  155616                    CLR     MMPRES          ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
```

INITIALIZE SECTION

```
4629 024564  005067  155614                    CLR     MMENAB          ;INDICATE MEM MGT IS NOT ENABLED.
4630 024570  004767  173330                    JSR     PC,CKTRAP       ;CLEAR THE MEM MGT SR0 REG AND CHECK FOR TRAP.
4631 024574  016767  155544  153202            MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4632 024602  103003                            BCC     10$             ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
4633 024604  012767  000001  155570            MOV     #1,MMPRES       ;INDICATE THAT MEM MGT IS PRESENT.
4634 024612  005067  155514          10$:      CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4635 024616  000167  000006                    JMP     NEWPAS          ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
4636
4637 024622                          NEWRES:   BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
     024622  104433                                                                  TRAP    C$RESET
4638 024624  005067  155502                    CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4639 024630                          NEWPAS:
4640 024630  012767  177777  155402            MOV     #-1,UNITN       ;RESET LOGICAL DEVICE TO -1
4641                                 ;*
4642                                 ; INCREMENT THE PASS COUNTER, CORRECT FOR ANY OVERFLOW.
4643                                 ;  THIS COUNTER IS USED IN THE ROM VERSION TEST.
4644                                 ;-
4645 024636  005267  155470                     INC     PASCNT          ;INCREMENT THE PASS COUNTER.
4646 024642  001002                            BNE     GETPRM          ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
4647 024644  005367  155462                     DEC     PASCNT          ;SET PASS COUNT TO 177777 OCTAL.
4648
4649                                 ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
4650 024650                          GETPRM:
4651 024650  005267  155364                    INC     UNITN           ;INCREMENT LOGICAL DEVICE NUMBER
4652 024654  026767  155360  155130            CMP     UNITN,L$UNIT    ;SEE IF MAXIMUM UNIT NO. EXCEEDED
4653 024662  002362                            BGE     NEWPAS          ;BR IF YES
4654
4655 024664                                    GPHARD  UNITN,R1        ;GET P-TABLE POINTER INTO R1
     024664  016700  155350                                                          MOV     UNITN,R0
     024670  104442                                                                  TRAP    C$GPHRD
     024672  010001                                                                  MOV     R0,R1
4656 024674                                    BCOMPLETE       30$     ;BR IF DEVICE AVAILABLE
     024674  103401                                                                  BCS     30$
4657 024676  000764                            BR      GETPRM          ;SKIP THIS DEVICE
4658
4659
4660                                 ;************ HARDWARE PARAMETER MOVING CODE ****************
4661 024700  012167  155340          30$:      MOV     (R1)+,CSRA      ;STORE DHV-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
4662 024704  012102                            MOV     (R1)+,R2        ;GET THE RX INTERRUPT VECTOR ADDRESS.
4663 024706  010267  155320                    MOV     R2,RXVECA       ;STORE RX INT VECTOR ADDRESS.
4664 024712  062702  000004                    ADD     #4,R2           ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
4665 024716  010267  155312                    MOV     R2,TXVECA       ;STORE TX INT VECTOR ADDRESS.
4666 024722  012167  155310                    MOV     (R1)+,ACTLNS    ;STORE DHV-11 ACTIVE LINE BIT MAP
4667 024726  012702  000377                    MOV     #MAPLNS,R2      ;GET THE BIT MAP FOR ALL LINES.
4668 024732  005102                            COM     R2              ;GET A BIT MAP OF NON-EXISTANT LINES.
4669 024734  040267  155276                    BIC     R2,ACTLNS       ;CLEAR NON-EXISTANT LINES FROM ACTLNS.
4670 024740  112167  155276                    MOVB    (R1)+,BRLEVL    ;STORE DHV-11 INTERUPT BUS REQUEST LEVEL
4671                                 ;*
4672                                 ;       CALCULTATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
4673                                 ;       DEVICE REGISTER ADDRESS TABLE.
4674                                 ;-
4675 024744  016701  155274                    MOV     CSRA,R1         ;COPY CSR ADDRESS
4676 024750  005201                            INC     R1              ;INCREMENT CSR ADDRESS
4677 024752  005201                            INC     R1              ; COPY BY 2.
4678 024754  012703  000007                    MOV     #7,R3           ;SET UP REGISTER COUNT
4679 024760  012702  002246                    MOV     #RBUFA,R2       ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
4680 024764  010122                   12$:     MOV     R1,(R2)+        ;STORE REGISTER ADDRESS IN TABLE
```

INITIALIZE SECTION

```
4681 024766  005201                          INC     R1              ;INCREMENT REGISTER ADDRESS
4682 024770  005201                          INC     R1              ; BY 2,FOR THE NEXT DEVICE REGISTER.
4683 024772  005303                          DEC     R3              ;DECREMENT REGISTER COUNT
4684 024774  001373                          BNE     12$             ;LOOP IF NOT DONE
4685
4686                                 ;+
4687                                 ; INITIALISE THE BMP CODE QUEUE.
4688                                 ;-
4689 024776  012700  002526                   MOV     #BMPCQB,R0      ;GET THE START ADDRESS OF THE QUEUE.
4690 025002  012701  002726                   MOV     #BMPCQE,R1      ;GET THE END ADDRESS OF THE QUEUE.
4691 025006  010067  155512                   MOV     R0,BMPCQP       ;SET THE POINTER TO THE START OF THE QUEUE.
4692 025012  005020                  14$:     CLR     (R0)+           ;CLEAR OUT THE CONTENTS OF THE QUEUE.
4693 025014  020001                          CMP     R0,R1           ;CHECK IF END OF QUEUE HAS BEEN REACHED.
4694 025016  103775                          BLO     14$             ;LOOP IF NOT ALL DONE.
4695                                 ;+
4696                                 ; REPORT THE UNIT NUMBER IF THE SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
4697                                 ;  AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
4698                                 ;-
4699 025020  032767  000020  155200           BIT     #BIT4,OPTION    ;CHECK IF THE QUESTION WAS ANSWERED YES.
4700 025026  001416                          BEQ     16$             ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
4701 025030  026727  154756  000001           CMP     L$UNIT,#1       ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
4702 025036  003412                          BLE     16$             ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
4703 025040                                  PRINTF  #MFUNIT,UNITN   ;REPORT UNIT NUMBER.
     025040  016746  155174                                                  MOV     UNITN,-(SP)
     025044  012746  004074                                                  MOV     #MFUNIT,-(SP)
     025050  012746  000002                                                  MOV     #2,-(SP)
     025054  010600                                                          MOV     SP,R0
     025056  104417                                                          TRAP    C$PNTF
     025060  062706  000006                                                  ADD     #6,SP
4704 025064                          16$:
4705
4706 025064  005067  155262          ENDIT:  CLR     CTRLCF          ;CLR THE CTRL-C TEST ABORT FLAG.
4707                                 ;+
4708                                 ; SET THE PROCESSOR PRIORITY TO ALLOW LTC INTERRUPTS BUT NOT OTHERS.
4709                                 ;-
4710 025070                                  SETPRI  #PRIO7                  ;SET PROCESSOR PRIORITY TO 7.
     025070  012700  000340                                                  MOV     #PRIO7,R0
     025074  104441                                                          TRAP    C$SPRI
4711 025076                                  ENDINIT
     025076                                                          L10021:
     025076  104411                                                          TRAP    C$INIT
4712
4713         000000                  TNUM == 0               ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.
```

INITIALIZE SECTION

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

025100                                  BGNAUTO
025100                                                                  L$AUTO::

025100                                  ENDAUTO
025100                                                                  L10022:
025100  104461                                                                    TRAP    C$AUTO
```

```
                        ;*****************************************************************************
                        ;
                        ;               FVT.CUC
                        ;
                        ;*****************************************************************************


                        .SBTTL  CLEANUP CODING SECTION

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

025102                          BGNCLN
025102                                                                  L$CLEAN::

005767  155244                   TST    CTRLCF          ;DID WE GET HERE BY CTRL-C FROM TEST?
001401                          BEQ     2$              ;CTRL-C FROM TEST?  NO, SKIP BUS RESET.
025110                          BRESET                  ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
025110  104433                                                          TRAP    C$RESET
025112                  2$:
025112                          EXIT    CLN
025112  104432                                                          TRAP    C$EXIT
025114  000002                                                          .WORD   L10023-.


                                .EVEN

025116                          ENDCLN
025116                                                          L10023:
025116  104412                                                          TRAP    C$CLEAN
```

CLEANUP CODING SECTION

```
4793
4794                            ;*****************************************************************************************
4795                            ;
4796                            ;               FVTA.DRP
4797                            ;
4798                            ;*****************************************************************************************
4799
4800
4801
4802                            .SBTTL  DROP UNIT SECTION
4803
4804                            ;++
4805                            ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4806                            ; TO NO LONGER BE TESTED.
4807                            ;--
4808
4809 025120                            BGNDU
       025120                                                                   L$DU::
4810
4811                            ;*****************************************************************************************
4812                            ;       INSERT DROP CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
4813                            ;       A "DROP" COMMAND OR A "DODU" MACRO EXECUTION.  THE PURPOSE
4814                            ;       OF THIS CODE IS TO DO ANY NECESSARY HOUSEKEEPING AFTER A
4815                            ;       UNIT HAS BEEN DROPPED.  THIS SECTION IS OPTIONAL.
4816                            ;*****************************************************************************************
4817
4818 025120                             PRINTF @DROP,R0        ;REPORT UNIT THAT HAS BEEN DROPPED.
       025120  010046                                                           MOV     R0,-(SP)
       025122  012746  025144                                                   MOV     @DROP,-(SP)
       025126  012746  000002                                                   MOV     #2,-(SP)
       025132  010600                                                           MOV     SP,R0
       025134  104417                                                           TRAP    C$PNTF
       025136  062706  000006                                                   ADD     #6,SP
4819 025142  000427                             BR      EDROP          ;BRANCH AROUND THE MESSAGE.
4820
4821 025144     045     101     040  DROP:     .ASCIZ/%A UNIT%D6%A DROPPED FROM FURTHER TESTING.%N/
       025147     125     116     111
       025152     124     045     104
       025155     066     045     101
       025160     040     104     122
       025163     117     120     120
       025166     105     104     040
       025171     106     122     117
       025174     115     040     106
       025177     125     122     124
       025202     110     105     122
       025205     040     124     105
       025210     123     124     111
       025213     116     107     056
       025216     045     116     000
4822                            .EVEN
4823 025222                     EDROP:
4824 025222                             EXIT    DU
       025222  000167                                                           .WORD   J$JMP
       025224  000000                                                           .WORD   L10024-2-.
4825
4826
```

DROP UNIT SECTION

```
4827
4828 025226                            E*NDDU
     025226                                                  L10024:
     025226  104453                                                 TRAP    C$DU
```

DROP UNIT SECTION

```
4830
4831                            ;*****************************************************************************
4832                            ;
4833                            ;               FVTA.ADD
4834                            ;
4835                            ;*****************************************************************************
4836
4837
4838
4839                            .SBTTL  ADD UNIT SECTION
4840
4841                            ;++
4842                            ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
4843                            ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
4844                            ;  TO THE TEST CYCLE.
4845                            ;--
4846
4847 025230                             BGNAU
    025230                                                              L$AU::
4848
4849                            ;*****************************************************************************
4850                            ;       INSERT ADD CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
4851                            ;       AN "ADD" COMMAND.  THE PURPOSE OF THIS CODE IS TO DO ANY
4852                            ;       HOUSEKEEPING THAT MAY BE NECESSARY AFTER A UNIT HAS BEEN ADDED.
4853                            ;       THIS SECTION IS OPTIONAL.
4854                            ;*****************************************************************************
4855
4856 025230                             EXIT    AU
    025230  000167                                                              .WORD   J$JMP
    025232  000000                                                              .WORD   L10025-2-.
4857
4858
4859                                    .EVEN
4860
4861 025234                             ENDAU
    025234                                                              L10025:
    025234  104452                                                              TRAP    C$AU
4862
```

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

025236                                  BGNTST
025236                                                                  T1::
000001                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025236  012767  000001  155062   MOV    @TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (1)
025244  012767  177777  155100   MOV    @-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
                                ;+
                                ; SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
                                ;-
025252  016767  152526  155064          MOV    4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
025260  012767  024134  152516          MOV    @TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
025266  005005                          CLR    R5              ;CLEAR THE ERROR FLAGS.

                                ;+
                                ; SET UP FOR THE INITIAL ITERATION OF THE TEST LOOP:
                                ;-
025270  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.
                                ;-
025272  005067  155050          2$:     CLR    TP4FLG          ;CLEAR THE 004 TRAP FLAG.
025276  016700  154742                  MOV    CSRA,R0         ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
025302  012701  025516                  MOV    @52$,R1         ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
025306  004767  172612                  JSR    PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
025312  103402                          BCS    4$              ;IF NO TRAP, BYPASS ERROR.
025314  052705  100001                  BIS    @100001,R5      ;SET FATAL READ ERROR FLAGS.
025320  042767  000017  000170  4$:     BIC    @17,52$         ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
025326  050467  000164                  BIS    R4,52$          ;OR IN THE LINE COUNTER TO THE I.A.R FIELD.
025332  010100                          MOV    R1,R0           ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
025334  016701  154704                  MOV    CSRA,R1         ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
025340  004767  172560                  JSR    PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
025344  103403                          BCS    6$              ;IF NO TRAP, BYPASS ERROR.
025346  052705  100002                  BIS    @100002,R5      ;SET FATAL WRITE ERROR FLAGS.
025352  000440                          BR     40$             ;EXIT AND REPORT FATAL ERROR.
                                ;+
                                ;    NOW, WE TEST EACH REGISTER FOR THIS LINE.
                                ;-
025354  012702  000010          6$:     MOV    @10,R2          ;INIT REGISTER COUNTER TO 8.
025360  016767  154660  000126          MOV    CSRA,50$                ;INITIALIZE THE REGISTER POINTER.
025366  012700  025514          8$:     MOV    @50$,R0         ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
025372  012701  025516                  MOV    @52$,R1         ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
025376  004767  172522                  JSR    PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
```

```
 4920 025402  103402                    BCS     10$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
 4921 025404  052705  100001             BIS     #100001,R5      ;SET FATAL READ ERROR FLAGS.
 4922 025410  010100               10$:  MOV     R1,R0           ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
 4923 025412  012701  025514             MOV     #50$,R1         ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.
 4924 025416  004767  172502             JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
 4925 025422  103402                    BCS     12$             ;IF NO TRAP  BYPASS THE SETTING OF ERROR FLAGS.
 4926 025424  052705  100002             BIS     #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
 4927 025430  005267  000060       12$:  INC     50$             ;INCREMENT THE REGISTER
 4928 025434  005267  000054             INC     50$             ; POINTER BY 2.
 4929 025440  005302                    DEC     R2              ;COUNT THE REGISTER.
 4930 025442  001351                    BNE     8$              ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
 4931
 4932                               ;+
 4933                               ; NOW WE SET UP TO TEST THE NEXT LINE, OR TO EXIT IF WE ARE DONE.
 4934                               ;-
 4935 025444  005204                    INC     R4              ;INCREMENT THE LINE COUNTER.
 4936 025446  020427  000010             CMP     R4,#NUMLNS      ;COMPARE LINE COUNTER AGAINST NUMBER OF LINES.
 4937 025452  002707                    BLT     2$              ;LOOP TO TEST THE NEXT LINE IF WE'RE NOT DONE.
 4938
 4939                               ;+
 4940                               ; DONE CHECKING DEVICE REGISTER ADDRESSES.
 4941                               ;    REPORT ANY ERRORS AND EXIT.
 4942                               ;-
 4943 025454  016767  154664  152322 40$:    MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
 4944 025462  005705                    TST     R5              ;CHECK THE ERROR FLAGS.
 4945 025464  100015                    BPL     60$             ;EXIT ROUTINE IF NO ERRORS.
 4946                                ; REPORT "DEVICE REGISTER ACCESS ERRORS"
 4947 025466                            ERRDF   101,EM0103,ER0101;              >>>>> ERROR #101 <<<<<.
      025466  104455                                                                  TRAP    C$ERDF
      025470  000145                                                                  .WORD   101
      025472  006065                                                                  .WORD   EM0103
      025474  016066                                                                  .WORD   ER0101
 4948
 4949 025476                            DODU    UNITN           ;DROP THIS UNIT FROM FUTHER TESTING.
      025476  016700  154536                                                          MOV     UNITN,R0
      025502  104451                                                                  TRAP    C$DODU
 4950 025504  005067  154642             CLR     CTRLCF          ;INDICATE NO CTRL-C ABORT FROM TEST.
 4951 025510                            DOCLN                   ;ABORT THIS SUB PASS.
      025510  104444                                                                  TRAP    C$DCLN
 4952 025512  000402                    BR      60$             ;
 4953                               ;+
 4954                               ; LOCAL STORAGE.
 4955                               ;-
 4956 025514  000000               50$:  .WORD   0       ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 4957 025516  000000               52$:  .WORD   0       ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 4958 025520  005067  154626       60$:   CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 4959 025524                            ENDTST
      025524                                                                  L10026:
      025524  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - MRSTA -
                                ;++ ************************************************************************
                                ;*              - MASTER RESET WITH SELFTEST TEST -
                                ;*      THIS TEST VERIFIES THAT THE MASTER RESET BIT WILL CLEAR AFTER A DEVICE
                                ;*      RESET AND THE PERFORMANCE OF THE DUT ROM BASED SELFTEST.
                                ;*
                                ;-- ************************************************************************
                                        BGNTST
                                                                                        T2::
                                        TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025526  012767  000002  154572    MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (2)
025534  012767  177777  154610    MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
025542                          SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
025542  012700  000240                                                          MOV     #PRIO5,R0
025546  104441                                                                  TRAP    C$SPRI
025550  012767  000001  156210    MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
025556  012767  006123  156206    MOV     #EM0201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
025564  012767  016404  156202    MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;-
025572  012701  011610            MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
025576  012702  000040            MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
025602  005003                    CLR     R3              ;WAITING FOR BIT TO CLEAR.
025604  016704  154434            MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
025610  004767  172522            JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
025614  103410                    BCS     2$              ;SKIP TO RESET DUT IF MR CLEAR.
                                ;+
                                ; DUT MASTER RESET BIT DID NOT GO CLEAR.  DEVICE MAY BE STUCK IN SOME
                                ;  ODD STATE.  TRY TO RESET DEVICE WITH A BUS RESET.
                                ;-
025616                              BRESET              ;NO, TRY TO JOG DEVICE WITH BUS RESET.
025616  104433                                                                  TRAP    C$RESET
025620  004767  174510            JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
025624  012701  011610            MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
025630  004767  172502            JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
025634  103016                    BCC     4$              ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
                                ;+
                                ; SET THE MASTER RESET BIT AND VERIFY THAT IT CLEARS WITHIN THE PROPER TIME.
                                ;-
025636  012701  011610    2$:     MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
025642  010214                    MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
025644  004767  172466            JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
025650  103010                    BCC     4$              ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
025652  012702  011610            MOV     #5000.,R2
025656  160102                    SUB     R1,R2           ;CALCULATE # OF MS FOR MR TO CLEAR.
025660  001413                    BEQ     6$              ;GO REPORT ERROR IF MR CLEAR IMMEDIATELY.
025662  020227  000764            CMP     R2,#500.
025666  002417                    BLT     8$              ;GO REPORT ERROR IF MR CLEAR IN < 1/2 SECOND.
025670  000424                    BR      60$             ;EXIT THE TEST WITHOUT ERROR.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
025672  012767  000311  156070  4$:     MOV     #201.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
025700  012701  006171                MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
025704                              ERROR               ;REPORT ERROR.          >>>>> ERROR #201 <<<<<
```

HARDWARE TEST - MRSTA -

```
        025704  104460                                                          TRAP    C$ERROR
5014    025706  000415                  BR      60$             ;EXIT THE TEST.
5015
5016                                    ;REPORT MR BIT CLEAR IMMEDIATELY AFTER DUT RESET.
5017    025710  012767  000312  156052  6$:     MOV     #202.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
5018    025716  012701  006344          MOV     #EM0203,R1      ;SELECT ERROR MESSAGE.
5019    025722                          ERROR                   ;REPORT ERROR.          >>>>> ERROR #202 <<<<<
        025722  104460                                                          TRAP    C$ERROR
5020    025724  000406                  BR      60$             ;EXIT THE TEST.
5021
5022                                    ;REPORT MR CLEAR WITHIN 1/2 SECOND OF DUT RESET.
5023    025726  012767  000313  156034  8$:     MOV     #203.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
5024    025734  012701  006507          MOV     #EM0204,R1      ;SELECT ERROR MESSAGE.
5025    025740                          ERROR                   ;REPORT ERROR.          >>>>> ERROR #203 <<<<<
        025740  104460                                                          TRAP    C$ERROR
5026
5027    025742                  60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
        025742  012700  000340                                                  MOV     #PRI07,R0
        025746  104441                                                          TRAP    C$SPRI
5028    025750  005067  154376          CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5029    025754                          ENDTST
        025754                                                          L10027:
        025754  104401                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - MRSSTA -
                                ;** ************************************************************************
                                ;*                  - MASTER RESET WITH SKIP SELFTEST TEST -
                                ;*      THIS TEST VERIFIES THAT THE MASTER RESET BIT WILL CLEAR AFTER A DEVICE
                                ;*      RESET AND THE SKIPPING OF THE DUT ROM BASED SELFTEST.
                                ;*
                                ;-- ************************************************************************
                                        BGNTST
                                                                                        T3::
                                        TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 025756  012767  000003  154342          MOV     @TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (3)
 025764  012767  177777  154360          MOV     @-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 025772                                 SETPRI  @PRI05          ;ALLOW LTC INTERRUPTS.
 025772  012700  000240                                                          MOV     @PRI05,R0
 025776  104441                                                                  TRAP    C$SPRI
 026000  012767  000001  155760          MOV     @1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 026006  012767  006666  155756          MOV     @EM0301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
 026014  012767  016404  155752          MOV     @ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;*
                                ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;-
 026022  012701  011610                 MOV     @5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
 026026  012702  000040                 MOV     @BIT05,R2       ;WAITING FOR MASTER RESET BIT.
 026032  005003                         CLR     R3              ;WAITING FOR BIT TO CLEAR.
 026034  016704  154204                 MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
 026040  004767  172272                 JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
 026044  103410                         BCS     2$              ;SKIP TO RESET DUT IF MR CLEAR.
                                ;*
                                ; DUT MASTER RESET BIT DID NOT GO CLEAR.  DEVICE MAY BE STUCK IN SOME
                                ;  ODD STATE.  TRY TO RESET DEVICE WITH A BUS RESET.
                                ;-
 026046                                 BRESET                  ;NO, TRY TO JOG DEVICE WITH BUS RESET.
 026046  104433                                                                 TRAP    C$RESET
 026050  004767  174260                 JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
 026054  012701  011610                 MOV     @5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
 026060  004767  172252                 JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
 026064  103024                         BCC     6$              ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
                                ;*
                                ; SET THE MASTER RESET BIT, TRY TO SKIP THE SELFTEST, AND VERIFY THAT THE
                                ;  MR BIT CLEARS WITHIN 1/5 SECOND.
                                ;-
 026066  012701  000310         2$:     MOV     @200.,R1        ;TIME-OUT VALUE IS 1/5 SECOND.
 026072  010214                         MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
 026074  004767  174234                 JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
 026100  004767  172232                 JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
 026104  103007                         BCC     4$              ;GO FIND OUT WHAT IS WRONG IF MR NOT CLEAR.
 026106  012702  000310                 MOV     @200.,R2
 026112  160102                         SUB     R1,R2           ;CALCULATE # OF MS FOR MR TO CLEAR.
 026114  020227  000012                 CMP     R2,@10.
 026120  002415                         BLT     8$              ;GO REPORT ERROR IF MR CLEAR IN < 10 MS.
 026122  000431                         BR      60$             ;EXIT THE TEST WITHOUT ERROR.
                                ;*
                                ; MR DID NOT CLEAR WITHIN 1/5 SECOND, SEE IF IT CLEARS WITHIN 5 SECONDS.
                                ;-
 026124  012701  011300         4$:     MOV     @4800.,R1       ;TIME-OUT VALUE IS 5 SECONDS MINUS 1/5 SECOND.
 026130  004767  172202                 JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
 026134  103416                         BCS     10$             ;GO REPORT ERROR IF MR CLEARED FINALLY.
```

```
5084                                    ;*
5085                                    ; ERROR REPORTS:
5086                                    ;-
5087                                        ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5088 026136  012767  000455  155624  6$:       MOV     #0301.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5089 026144  012701  006171                   MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
5090 026150                                   ERROR                   ;REPORT ERROR.          >>>>> ERROR #0301 <<<<<
     026150  104460                                                                              TRAP    C$ERROR
5091 026152  000415                           BR      60$             ;EXIT THE TEST.
5092
5093                                        ;REPORT MR BIT CLEAR WITHIN 10 MS AFTER DUT RESET.
5094 026154  012767  000456  155606  8$:       MOV     #0302.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5095 026162  012701  006731                   MOV     #EM0302,R1      ;SELECT ERROR MESSAGE.
5096 026166                                   ERROR                   ;REPORT ERROR.          >>>>> ERROR #0302 <<<<<
     026166  104460                                                                              TRAP    C$ERROR
5097 026170  000406                           BR      60$             ;EXIT THE TEST.
5098
5099                                        ;REPORT MR CLEARED BETWEEN 1/5 SECOND AND 5 SECONDS OF DUT RESET.
5100 026172  012767  000457  155570  10$:      MOV     #0303.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5101 026200  012701  007071                   MOV     #EM0303,R1      ;SELECT ERROR MESSAGE.
5102 026204                                   ERROR                   ;REPORT ERROR.          >>>>> ERROR #0303 <<<<<
     026204  104460                                                                              TRAP    C$ERROR
5103
5104 026206                            60$:   SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     026206  012700  000340                                                                      MOV     #PRI07,R0
     026212  104441                                                                              TRAP    C$SPRI
5105 026214  005067  154132                    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5106 026220                                   ENDTST
     026220                                                                             L10030:
     026220  104401                                                                              TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RXCHRA -
                                ;** ****************************************************************************
                                ;*              - RBUF REGISTER RX CHARACTER FIELD TEST -
                                ;*      THIS TEST VERIFIES THAT THE RX CHARACTER FIELD OF THE DUT RBUF REGISTER
                                ;*      APPEARS TO BE FUNCTIONING CORRECTLY.  THIS TEST USES THE CODES WHICH
                                ;*      SHOULD BE IN THE FIFO AFTER A BOARD RESET AND SKIP SELFTEST SEQUENCE.
                                ;*
                                ;-- ****************************************************************************
                                        BGNTST
                                                                                T4::
                                        SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
026222  012700  000240                                                    MOV     #PRI05,R0
026226  104441                                                          TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
000004
026230  012767  000004  154070         MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (4)
026236  012767  177777  154106         MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
026244  012767  000001  155514         MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026252  012767  007250  155512         MOV     #EM0401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026260  012767  016404  155506         MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
026266  012701  011610                 MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
026272  012702  000040                 MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
026276  005003                         CLR     R3              ;WAITING FOR BIT TO CLEAR.
026300  016704  153740                 MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
026304  010214                         MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
026306  004767  174022                 JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
026312  004767  172020                 JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026316  103015                         BCC     4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; READ 6 CHARACTERS FROM THE DUT AND VERIFY THAT THEY ARE VALID SELFTEST
                                ;  CODES.
                                ;-
026320  012400                          MOV     (R4)+,R0        ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
026322  012701  000006                  MOV     #6,R1           ;INITIALIZE THE LOOP COUNTER.
026326  011402                 2$:      MOV     (R4),R2         ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
026330  010200                          MOV     R2,R0
026332  042700  177476                  BIC     #177476,R0      ;REMOVE ALL BUT BITS SPECIFIC TO SELFTEST CODE.
026336  020027  000201                  CMP     R0,#201         ;CHECK THAT BITS 0,6, AND 7 ARE CORRECT.
026342  001012                         BNE     6$              ;GO REPORT ERROR IF CODE IS NOT SELFTEST CODE.
026344  005301                          DEC     R1              ;COUNT THIS LOOP ITERATION.
026346  001367                         BNE     2$              ;LOOP IF NOT ALL LINES DONE.
026350  000415                         BR      60$             ;EXIT TEST, NO ERROR FOUND.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
026352  012767  000621  155410  4$:       MOV     #0401.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE
026360  012701  006171                    MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
026364                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #0401 <<<<<
026364  104460                                                                            TRAP    C$ERROR
026366  000406                            BR      60$             ;EXIT THE TEST.

                                    ;REPORT IMPROPER CODE FOUND IN DUT RBUF AFTER RESET (SKIP SELFTEST).
```

HARDWARE TEST - RXCHRA -

```
5161 026370  012767  000622  155372  6$:    MOV     #0402.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5162 026376  012701  007317                 MOV     #EM0402,R1      ;SELECT ERROR MESSAGE.
5163 026402                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0402 <<<<<
     026402  104460                                                                              TRAP    C$ERROR
5164
5165 026404                         60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     026404  012700  000340                                                                      MOV     #PRI07,R0
     026410  104441                                                                              TRAP    C$SPRI
5166 026412  005067  153734                 CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5167 026416                                 ENDTST
     026416                                                                         L10031:
     026416  104401                                                                              TRAP    C$ETST
```

HARDWARE TEST - RXFFDA -

```
                                .SBTTL  HARDWARE TEST           - RXFFDA -
                                ;** ****************************************************************************
                                ;*              - RBUF REGISTER RX FLAG FIELD TEST -
                                ;*      THIS TEST VERIFIES THAT THE FIELD OF 3 FLAG BITS IN THE RBUF READS
                                ;*      AS ALL ONES WHEN THE SELFTEST CODES ARE BEING READ FROM THE DUT
                                ;*      AFTER A BOARD RESET AND SKIP SELFTEST SEQUENCE.
                                ;*
                                ;-- ****************************************************************************
026420                                  BGNTST
026420                                                                          T5::
026420                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
026420  012700  000240                                                        MOV     #PRIO5,R0
026424  104441                                                                TRAP    C$SPRI
        000005                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026426  012767  000005  153672          MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (5)
026434  012767  177777  153710          MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
026442  012767  000001  155316          MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026450  012767  007467  155314          MOV     #EM0501,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026456  012767  016404  155310          MOV     #ER0201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
026464  012701  011610                  MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
026470  012702  000040                  MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
026474  005003                          CLR     R3             ;WAITING FOR BIT TO CLEAR.
026476  016704  153542                  MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
026502  010214                          MOV     R2,(R4)        ;SET THE DUT MASTER RESET BIT.
026504  004767  173624                  JSR     PC,SKPSTS      ;SKIP THE SELFTEST.
026510  004767  171622                  JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026514  103013                          BCC     4$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; READ 8 CHARACTERS FROM THE DUT AND VERIFY THAT ALL 3 RX ERROR FLAGS ARE
                                ;  SET FOR EACH CHARACTERS.
                                ;-
026516  012400                          MOV     (R4)+,R0       ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
026520  012701  000010                  MOV     #8.,R1         ;INITIALIZE THE LOOP COUNTER.
026524  011402                  2$:     MOV     (R4),R2        ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
026526  012700  070000                  MOV     #70000,R0
026532  040200                          BIC     R2,R0          ;CALCULATE BIT MAP OF CLEAR RX ERROR FLAGS.
026534  001012                          BNE     6$             ;GO REPORT ERROR IF NOT ALL RX ERROR FLAGS SET.
026536  005301                          DEC     R1             ;COUNT THIS LOOP ITERATION.
026540  001371                          BNE     2$             ;LOOP IF NOT ALL LINES DONE.
026542  000415                          BR      60$            ;EXIT TEST, NO ERROR FOUND.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
026544  012767  000765  155216  4$:      MOV     #0501.,ERRNBR  ;SET THE ERROR NUMBER IN ERROR TABLE.
026552  012701  006171                  MOV     #EM0202,R1     ;SELECT ERROR MESSAGE.
026556                                  ERROR                  ;REPORT ERROR.          >>>>> ERROR #0501 <<<<<
026556  104460                                                                          TRAP    C$ERROR
026560  000406                          BR      60$            ;EXIT THE TEST.
                                    ;REPORT ONE OR MORE RX ERROR FLAGS FOUND SET WITH SELFTEST CODE.
026562  012767  000766  155200  6$:      MOV     #0502.,ERRNBR  ;SET THE ERROR NUMBER IN ERROR TABLE.
```

```
HARDWARE TEST            - RXFFDA -

   5222 026570  012701  007535                MOV     #EM0502,R1      ;SELECT ERROR MESSAGE.
   5223 026574                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0502 <<<<<
        026574  104460                                                                               TRAP    C$ERROR
   5224
   5225 026576                          60$:   SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
        026576  012700  000340                                                                       MOV     #PRI07,R0
        026602  104441                                                                               TRAP    C$SPRI
   5226 026604  005067  153542                 CLR     CTRLCF         ;INDICATE THAT WE COMPLETED THE TEST.
   5227 026610                                 ENDTST
        026610                                                                          L10032:
        026610  104401                                                                               TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RDAA -
                                ;** ************************************************************************
                                ;*              - CSR RX DATA AVAILABLE BIT TEST -
                                ;*      THIS TEST VERIFIES THAT THE DUT CSR RX DATA AVAILABLE BIT IS SET BY THE
                                ;*      INCLUSION OF THE SELFTEST CODES IN THE DUT FIFO AND THAT THE BIT CLEARS
                                ;*      AFTER THE FIFO HAS BEEN EMPTIED.
                                ;*
                                ;-- ************************************************************************
026612                                  BGNTST
026612                                                                          T6::
026612                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
026612  012700  000240                                                        MOV     #PRI05,R0
026616  104441                                                                TRAP    C$SPRI
        000006                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026620  012767  000006  153500           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (6)
026626  012767  177777  153516           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
026634  012767  000001  155124           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026642  012767  010071  155122           MOV    #EM0601,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026650  012767  016404  155116           MOV    #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
026656  012701  011610                   MOV    #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
026662  012702  000040                   MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
026666  005003                           CLR    R3              ;WAITING FOR BIT TO CLEAR.
026670  016704  153350                   MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
026674  010214                           MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
026676  004767  173432                   JSR    PC,SKPSTS       ;SKIP THE SELFTEST.
026702  004767  171430                   JSR    PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026706  103016                           BCC    4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; CHECK THAT THE RX DATA AVAILABLE BIT IS SET.
                                ;-
026710  032714  000200                   BIT    #BIT7,(R4)      ;TEST THE DUT RX.DATA.AVAIL BIT.
026714  001422                           BEQ    6$              ;GO REPORT ERROR IF BIT IS NOT SET.
                                ;+
                                ; READ CHARACTERS FROM THE DUT RX FIFO AND WAIT FOR RX.DATA.AVAIL TO GO CLEAR.
                                ;-
026716  012705  001130                   MOV    #600.,R5        ;ALLOW READING 600 CHARS BEFORE ERROR.
026722  010403                           MOV    R4,R3
026724  012300                           MOV    (R3)+,R0        ;CALCULATE THE RBUF ADDRESS.
026726  011300                   2$:     MOV    (R3),R0         ;READ A CHARACTER FROM THE RX FIFO.
026730  032714  000200                   BIT    #BIT7,(R4)      ;TEST THE DUT RX.DATA.AVAIL BIT.
026734  001427                           BEQ    60$             ;EXIT TEST WITHOUT ERROR IF RX.DATA.AVAIL CLR.
026736  005305                           DEC    R5              ;COUNT THE CHARACTER JUST READ.
026740  001372                           BNE    2$              ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
026742  000416                           BR     8$              ;GO REPORT ERROR IF RX.DATA.AVAIL WOULDN'T CLR.

                                ;+
                                ; ERROR REPORTS:
                                ;-
                                   ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
026744  012767  001131  155016   4$:     MOV    #0601.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026752  012701  006171                   MOV    #EM0202,R1      ;SELECT ERROR MESSAGE.
026756                                   ERROR                  ;REPORT ERROR.          >>>>> ERROR #0601 <<<<<
026756  104460                                                                          TRAP    C$ERROR
```

```
 5282 026760  000415                          BR      60$             ;EXIT THE TEST.
 5283
 5284                                ;REPORT THAT RX.DATA.AVAIL BIT WAS NOT SET AFTER A RESET COMPLETION.
 5285 026762  012767  001132  155000  6$:     MOV     #0602.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
 5286 026770  012701  010125                  MOV     #EM0602,R1      ;SELECT ERROR MESSAGE.
 5287 026774                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0602 <<<<<
      026774  104460                                                                          TRAP    C$ERROR
 5288 026776  000406                          BR      60$             ;EXIT THE TEST.
 5289
 5290                                ;REPORT THAT RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO.
 5291 027000  012767  001133  154762  8$:     MOV     #0603.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
 5292 027006  012701  010305                  MOV     #EM0603,R1      ;SELECT ERROR MESSAGE.
 5293 027012                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0603 <<<<<
      027012  104460                                                                          TRAP    C$ERROR
 5294
 5295 027014                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      027014  012700  000340                                                                  MOV     #PRI07,R0
      027020  104441                                                                          TRAP    C$SPRI
 5296 027022  005067  153324                   CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5297 027026                                  ENDTST
      027026                                                                      L10033:
      027026  104401                                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST            - RDVA -
                                ;++ ******************************************************************************
                                ;*              - RBUF RX DATA VALID BIT TEST -
                                ;*      THIS TEST VERIFIES THAT THE DUT RBUF RX DATA VALID BIT IS SET BY THE
                                ;*      INCLUSION OF THE SELFTEST CODES IN THE DUT FIFO AND THAT THE BIT CLEARS
                                ;*      AFTER THE FIFO HAS BEEN EMPTIED.
                                ;*
                                ;-- ******************************************************************************
                                        BGNTST
                                                                                        T7::
                                        SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
027030  012700  000240                                                          MOV     #PRIO5,R0
027034  104441                                                                 TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027036  012767  000007  153262         MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (7)
027044  012767  177777  153300         MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
027052  012767  000001  154706         MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027060  012767  010470  154704         MOV     #EM0701,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
027066  012767  016404  154700         MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
027074  012701  011610                 MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
027100  012702  000040                 MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
027104  005003                         CLR     R3              ;WAITING FOR BIT TO CLEAR.
027106  016704  153132                 MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
027112  010214                         MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
027114  004767  173214                 JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
027120  004767  171212                 JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
027124  103012                         BCC     4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; CHECK THAT THE RX DATA VALID BIT IS SET.
                                ;-
027126  012400                          MOV     (R4)+,R0        ;INCREMENT POINTER TO PNT TO DUT RBUF REG.
027130  005714                          TST     (R4)            ;TEST THE DUT RX.DATA.VALID BIT.
027132  100016                        BPL     6$              ;GO REPORT ERROR IF BIT IS NOT SET.
                                ;+
                                ; READ CHARACTERS FROM THE DUT RX FIFO AND WAIT FOR RX.DATA.VALID TO GO CLEAR.
                                ;-
027134  012705  001130                  MOV     #600.,R5        ;ALLOW READING 600 CHARS BEFORE ERROR.
027140  011400                 2$:      MOV     (R4),R0         ;READ A CHARACTER FROM THE RX FIFO.
027142  100027                        BPL     60$             ;EXIT TEST WITHOUT ERROR IF BIT IS CLEAR.
027144  005305                          DEC     R5              ;COUNT THE CHARACTER JUST READ.
027146  001374                        BNE     2$              ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
027150  000416                          BR      8$              ;GO REPORT ERROR IF RX.DATA.VALID WOULDN'T CLR.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
027152  012767  001275  154610  4$:      MOV     #0701.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
027160  012701  006171                 MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
027164                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0701 <<<<<
027164  104460                                                                   TRAP    C$ERROR
027166  000415                         BR      60$             ;EXIT THE TEST.
```

HARDWARE TEST - RDVA -

```
5352                                    ;REPORT THAT RX.DATA.VALID BIT WAS NOT SET AFTER A RESET COMPLETION.
5353 027170  012767  001276  154572  6$:     MOV     #0702.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5354 027176  012701  010525                  MOV     #EM0702,R1      ;SELECT ERROR MESSAGE.
5355 027202                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0702 <<<<<
     027202  104460                                                                              TRAP    C$ERROR
5356 027204  000406                          BR      60$             ;EXIT THE TEST.
5357
5358                                    ;REPORT THAT RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO.
5359 027206  012767  001277  154554  8$:     MOV     #0703.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5360 027214  012701  010705                  MOV     #EM0703,R1      ;SELECT ERROR MESSAGE.
5361 027220                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0703 <<<<<
     027220  104460                                                                              TRAP    C$ERROR
5362
5363 027222                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     027222  012700  000340                                                                      MOV     #PRI07,R0
     027226  104441                                                                              TRAP    C$SPRI
5364 027230  005067  153116                   CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5365 027234                                  ENDTST
     027234                                                                          L10034:
     027234  104401                                                                              TRAP    C$ETST
```

HARDWARE TEST - RLNA -

```
                                .SBTTL  HARDWARE TEST           - RLNA -
                                ;++ ************************************************************************
                                ;*              - RBUF RX LINE NUMBER FIELD TEST -
                                ;*      THIS TEST VERIFIES THAT THE DUT RBUF RX LINE NUMBER FIELD IS WORKING
                                ;*      CORRECTLY BY UTILIZING THE SELFTEST CODES WHICH ARE PUT IN THE RX
                                ;*      FIFO AFTER A BOARD RESET.
                                ;*
                                ;-- ************************************************************************
027236                          BGNTST
027236                                                                          T8::
027236                          SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
027236  012700  000240                                                        MOV     #PRIO5,R0
027242  104441                                                               TRAP    C$SPRI
        000010                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027244  012767  000010  153054    MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (8)
027252  012767  177777  153072    MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
027260  012767  000001  154500    MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027266  012767  011070  154476    MOV     #EM0801,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
027274  012701  011610          MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
027300  012702  000040          MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
027304  005003                  CLR     R3             ;WAITING FOR BIT TO CLEAR.
027306  016704  152732          MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
027312  010214                  MOV     R2,(R4)        ;SET THE DUT MASTER RESET BIT.
027314  004767  173014          JSR     PC,SKPSTS      ;SKIP THE SELFTEST.
027320  004767  171012          JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
027324  103016                  BCC     4$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; READ CHARACTERS FROM THE DUT RX FIFO AND VERIFY THAT THE LINE NUMBERS ARE
                                ;  CORRECT.
                                ; ONE CHARACTER IS READ FROM THE FIFO FOR EACH POSSIBLE LINE ON THE DUT.
                                ;-
027326  005001                  CLR     R1             ;CLEAR THE LINE COUNTER.
027330  012400                  MOV     (R4)+,R0       ;INCREMENT POINTER TO PNT TO THE DUT RBUF REG.
027332  011402          2$:     MOV     (R4),R2        ;READ A CHARACTER FROM THE DUT RX FIFO.
027334  010203                  MOV     R2,R3
027336  000303                  SWAB    R3
027340  042703  177760          BIC     #177760,R3     ;REMOVE ALL BUT LINE NUMBER BITS.
027344  020301                  CMP     R3,R1          ;COMPARE WITH EXPECTED LINE NUMBER.
027346  001017                  BNE     6$             ;GO REPORT ERROR IF LINE NUMBERS DON'T MATCH.
027350  005201                  INC     R1             ;INCREMENT THE EXPECTED LINE NUMBER.
027352  020127  000010          CMP     R1,#NUMLNS     ;COMPARE WITH NUMBER OF LINES ON DUT.
027356  001365                  BNE     2$             ;LOOP UNTIL CODES FOR ALL LINES ARE READ.
027360  000423                  BR      60$            ;EXIT TEST WITHOUT ERROR.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
027362  012767  001441  154400  4$:     MOV     #0801.,ERRNBR  ;SET THE ERROR NUMBER IN ERROR TABLE.
027370  012767  016466  154376          MOV     #ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
027376  012701  006171                  MOV     #EM0202,R1     ;SELECT ERROR MESSAGE.
027402                          ERROR                   ;REPORT ERROR.          >>>>> ERROR #0801 <<<<<
027402  104460                                                                  TRAP    C$ERROR
```

HARDWARE TEST - RLNA -

```
5420 027404  000411                          BR      60$             ;EXIT THE TEST.
5421
5422                                      ;REPORT THAT RX LINE NUMBER FIELD IS WRONG FOR SELFTEST CODE.
5423 027406  012767  001442  154354  6$:     MOV     #0802.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5424 027414  012767  016404  154352          MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5425 027422  012701  011130                  MOV     #EM0802,R1      ;SELECT ERROR MESSAGE.
5426 027426                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0802 <<<<<
     027426  104460                                                                              TRAP    C$ERROR
5427
5428 027430                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     027430  012700  000340                                                                      MOV     #PRI07,R0
     027434  104441                                                                              TRAP    C$SPRI
5429 027436  005067  152710                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5430 027442                                  ENDTST
     027442                                                                          L10035:
     027442  104401                                                                              TRAP    C$ETST
```

```
5432                                    .SBTTL  HARDWARE TEST            - BMPCHK -
5433                                    ;** ************************************************************************
5434                                    ;*              - BMP CHECK TEST -
5435                                    ;*      THIS TEST IS USED TO VERIFY THAT THE DUT DOES NOT IMMEDIATELY FAIL
5436                                    ;*      THE ON-BOARD BACKGROUND-MONITOR PROGRAM, AND HENCE INVALIDATE
5437                                    ;*      SUCCEEDING TESTS.
5438                                    ;*      THIS TEST LOOKS FOR BMP CODES IN THE FIFO FOR A SET PERIOD IMMEDIATELY
5439                                    ;*      AFTER THE SELF-TEST IS SKIPPED.
5440                                    ;*      ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE QUEUE AND ARE ALSO
5441                                    ;*      REPORTED IN THIS TEST.
5442                                    ;*
5443                                    ;-- ************************************************************************
5444 027444                                    BGNTST
      027444                                                                    T9::
5445 027444                                    SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
      027444  012700  000240                                                            MOV     #PRIO5,R0
      027450  104441                                                                    TRAP    C$SPRI
5446            000011                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5447 027452  012767  000011  152646            MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.                (9)
5448 027460  012767  177777  152664            MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
5449 027466  012767  000001  154272            MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5450 027474  012767  001605  154266            MOV     #0901.,ERRNBR   ;SET THE ERROR NUMBER.
5451                                    ;*
5452                                    ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5453                                    ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5454                                    ;-
5455 027502  012701  005670                    MOV     #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
5456 027506  012702  000040                    MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
5457 027512  005003                            CLR     R3              ;WAITING FOR BIT TO CLEAR.
5458 027514  016704  152524                    MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
5459 027520  004767  170612                    JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5460 027524  103027                            BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
5461                                    ;*
5462                                    ; RESET THE DUT, SKIP THE SELF-TEST.
5463                                    ;-
5464 027526  010214                            MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
5465 027530  004767  172600                    JSR     PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
5466                                    ;*
5467                                    ; WAIT FOR MASTER RESET TO CLEAR. DELAY FOR 500 MILLI-SECS BEFORE PURGING
5468                                    ; THE FIFO.
5469                                    ;-
5470 027534  012704  000764                    MOV     #500.,R4        ;TIME-OUT VALUE IS 500 MILLI-SECONDS.
5471 027540  004767  170532                    JSR     PC,DELAY        ;WAIT FOR BMP TO BEGIN EXECUTION.
5472 027544  004767  171144                    JSR     PC,PUFIFO       ;PURGE THE FIFO, SAVING ANY BMP CODES.
5473 027550  103015                            BCC     50$             ;ABORT THE TEST IF THE FIFO DID NOT CLEAR.
5474                                    ;*
5475                                    ; REPORT THE ERROR IF ANY BMP CODES WERE FOUND.
5476                                    ;-
5477 027552  016702  152746                    MOV     BMPCQP,R2       ;GET THE CONTENTS OF THE POINTER TO THE BMP Q.
5478 027556  012703  002526                    MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
5479 027562  020203                            CMP     R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
5480 027564  001414                            BEQ     60$             ;EXIT NO CODES IN THE QUEUE.
5481                                    ;*
5482                                    ; THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
5483                                    ;-
5484                                            ;REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN"
5485 027566  012701  011234                    MOV     #EM0902,R1      ;PASS THE MESSAGE TO BE REPORTED.
```

HARDWARE TEST - BMPCHK -

```
5486 027572                                    ERRDF   0901,EM0901,ER9301 ;          >>>>> ERROR #0901 <<<<<.
     027572  104455                                                                    TRAP    C$ERDF
     027574  001605                                                                    .WORD   901
     027576  011203                                                                    .WORD   EM0901
     027600  017446                                                                    .WORD   ER9301
5487 027602  000405                                    BR      60$
5488
5489 027604  012767  001606  154156  50$:    MOV     #902.,ERRNBR    ;SET >>>>> ERROR #0902 <<<<<.
5490 027612  004767  172644                   JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
5491
5492 027616                            60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     027616  012700  000340                                                           MOV     #PRI07,R0
     027622  104441                                                                   TRAP    C$SPRI
5493 027624  005067  152522                    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5494 027630                                    ENDTST
     027630                                                                 L10036:
     027630  104401                                                                   TRAP    C$ETST
```

```
5496
5497
5498                                    .SBTTL  HARDWARE TEST           - SKSELF -
5499                                    ;** ************************************************************************
5500                                    ;*              - SKIP SELF-TEST TEST -
5501                                    ;*      THIS TEST VERIFIES THAT THE DUT SKIPS THE SELF-TEST WITHIN THE
5502                                    ;*      TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS
5503                                    ;*      COMPLETION.
5504                                    ;*
5505                                    ;** ************************************************************************
5506 027632                                    BGNTST
     027632                                                                    T10::
5507 027632                                    SETPRI  @PRIO5          ;ALLOW LTC INTERRUPTS.
     027632  012700  000240                                                     MOV     @PRIO5,R0
     027636  104441                                                             TRAP    C$SPRI
5508            000012                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5509 027640  012767  000012  152460            MOV     @TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (10)
5510 027646  012767  177777  152476            MOV     @-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
5511 027654  012767  000001  154104            MOV     @1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5512 027662  012767  011270  154102            MOV     @EM1001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5513 027670  012767  016466  154076            MOV     @ERO503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5514                                    ;*
5515                                    ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5516                                    ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5517                                    ;-
5518 027676  012701  005670                    MOV     @3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
5519 027702  012702  000040                    MOV     @BIT05,R2       ;WAITING FOR MASTER RESET BIT.
5520 027706  005003                            CLR     R3              ;WAITING FOR BIT TO CLEAR.
5521 027710  016704  152330                    MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
5522 027714  004767  170416                    JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5523 027720  103037                            BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
5524                                    ;*
5525                                    ; DETERMINE IF THE DUT TAKES TOO SHORT OR TOO LONG A TIME TO SKIP THE SELF-TEST
5526                                    ;  SET-UP A TIME-OUT OF 50 MILLI-SECOND, IF MR IS CLEAR IN LESS THAN 10 MILLI
5527                                    ;    -SECOND, OR GREATER THAN 50 MILLI-SECONDS, REPORT THE ERROR.
5528                                    ;-
5529 027722  012701  000062                    MOV     @50.,R1         ;TIME-OUT VALUE IS 50 MILLI-SECONDS.
5530 027726  010214                            MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
5531 027730  004767  172400                    JSR     PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
5532 027734  004767  170376                    JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5533 027740  103011                            BCC     2$              ;GO REPORT ERR IF SKIPPING STEST TOOK TOO LONG.
5534 027742  020127  000050                    CMP     R1,@40.
5535 027746  003015                            BGT     4$              ;GO REP ERR IF SELFTEST COMPLETED IN < 10 MS.
5536                                    ;*
5537                                    ; SELF-TEST COMPLETED WITHIN 10 MILLI-SEC TO 50 MILLI-SECONDS.
5538                                    ;  VERIFY THAT THE SELF-TEST CODES IN THE FIFO ARE "GOOD" CODES ,IE THE DUT
5539                                    ;  SUCCESSFULLY COMPLETED THE SELF-TEST.
5540                                    ; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 1003 THRU 1007 <<<<<.
5541                                    ;-
5542 027750  012767  001753  154012            MOV     @1003.,ERRNBR   ;SET ERROR NUMBER TO 1003.
5543 027756  004767  171636                    JSR     PC,RSTRPT       ;CHECK SELF-TEST CODES IN THE FIFO.
5544 027762  000423                            BR      60$             ;EXIT TEST.
5545                                    ;*
5546                                    ; ERROR REPORTS:
5547                                    ;-
5548                                      ;REPORT SKIP SELF-TEST TOOK TOO LONG.
5549 027764  012767  001751  153776    2$:     MOV     @1001.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
```

HARDWARE TEST - SKSELF -

```
5550 027772  012701  011334                MOV     #EM1002,R1      ;SELECT ERROR MESSAGE.
5551 027776                                ERROR                   ;REPORT ERROR.          >>>>> ERROR #1001 <<<<<
     027776  104460                                                                              TRAP    C$ERROR
5552 030000  000414                        BR      60$             ;EXIT THE TEST.
5553
5554                                ;REPORT SKIP SELF-TEST COMPLETED TOO SOON.
5555 030002  012767  001752  153760  4$:   MOV     #1002.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROW TABLE.
5556 030010  012701  011421                MOV     #EM1003,R1      ;SELECT ERROR MESSAGE.
5557 030014                                ERROR                   ;REPORT ERROR.          >>>>> ERROR #1002 <<<<<
     030014  104460                                                                              TRAP    C$ERROR
5558 030016  000405                        BR      60$             ;EXIT THE TEST.
5559
5560 030020  012767  001753  153742  50$:   MOV     #1003.,ERRNBR   ;SET ERROR NUMBER.
5561 030026  004767  172430                JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
5562
5563 030032                          60$:   SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     030032  012700  000340                                                                      MOV     #PRI07,R0
     030036  104441                                                                              TRAP    C$SPRI
5564 030040  005067  152306                CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5565 030044                                ENDTST
     030044                                                                     L10037:
     030044  104401                                                                              TRAP    C$ETST
```

HARDWARE TEST - SKSELF -

```
5567
5568
5569                                    .SBTTL  HARDWARE TEST            - DFSKST -
5570                                    ;** ************************************************************************
5571                                    ;*              - DIAGNOSTIC FAIL BIT, SKIP SELF-TEST TEST -
5572                                    ;*      THIS TEST VERIFIES THAT THE DIAGNOSTIC FAIL BIT OF THE DUT, CORRECTLY
5573                                    ;*      CHANGES STATE AS THE ON-BOARED SELFTEST IS SKIPPED.
5574                                    ;*
5575                                    ;-- ************************************************************************
5576 030046                                 BGNTST
      030046                                                                          T11::
5577 030046                                 SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
      030046  012700  000240                                                        MOV     #PRIO5,R0
      030052  104441                                                                TRAP    C$SPRI
5578          000013                         TNUM = TNUM + 1         ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5579 030054  012767  000013  152244         MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (11)
5580 030062  012767  177777  152262         MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
5581 030070  012767  000001  153670         MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5582 030076  012767  011477  153666         MOV     #EM1101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5583 030104  012767  016466  153662         MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5584                                    ;*
5585                                    ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5586                                    ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5587                                    ;-
5588 030112  012701  005670                     MOV     #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
5589 030116  012702  000040                     MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
5590 030122  005003                             CLR     R3              ;WAITING FOR BIT TO CLEAR.
5591 030124  016704  152114                     MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
5592 030130  004767  170202                     JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5593 030134  103044                             BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
5594                                    ;*
5595                                    ; RESET THE DUT, SKIP THE SELF-TEST.
5596                                    ;-
5597 030136  010214                             MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
5598 030140  004767  172170                     JSR     PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
5599                                    ;*
5600                                    ; SET TIME OUT OF 5 MILLI SECONDS, WAIT FOR DIAG_FAIL BIT TO SET.
5601                                    ;  IF TIME-OUT OCCURS GO REPORT THE ERROR.
5602                                    ;-
5603 030144  012701  000005                     MOV     #5,R1           ;TIME-OUT VALUE IS 5 MILLI-SECONDS.
5604 030150  012702  020000                     MOV     #BIT13,R2       ;WAITING FOR DIAGNOSTIC FAIL BIT.
5605 030154  010203                             MOV     R2,R3           ;WAITING FOR BIT TO SET.
5606 030156  016704  152062                     MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
5607 030162  004767  170150                     JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
5608 030166  103020                             BCC     4$              ;IF DIAG_FAIL DID NOT SET, GO REPORT ERROR.
5609
5610                                    ;*
5611                                    ; SET TIME-OUT OF 15 MILLI-SECS, WAIT FOR DIAG_FAIL TO CLEAR.
5612                                    ;  IF TIME-OUT OCCURS GO REPORT THE ERROR.
5613                                    ;  VERIFY THE DIAG_FAIL BIT IS IN A STABLE STATE BEFORE CONTINUING.  LOOP
5614                                    ;  BACK IF THE STATE WAS TRANSITORY, USING THE REMAINDER OF THE 15 MS TIME-OUT.
5615                                    ;-
5616 030170  012701  000017                     MOV     #15.,R1         ;TIME-OUT VALUE IS 15 MILLI-SECONDS.
5617 030174  005003                     2$:     CLR     R3              ;WAITING FOR BIT TO CLEAR.
5618 030176  004767  170134                     JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
5619 030202  103012                             BCC     4$              ;IF DIAG_FAIL DID NOT CLEAR, GO REPORT ERROR.
5620 030204  010105                             MOV     R1,R5           ;SAVE THE REMAINING TIME-OUT VALUE.
```

HARDWARE TEST - DFSKST -

```
5621 030206  012701  000001                MOV    #1,R1           ;SET TIME-OUT OF 1 MILLI-SECOND.
5622 030212  052703  020000                BIS    #BIT13,R3       ;WAIT FOR BIT TO SET.
5623 030216  004767  170114                JSR    PC,MSLGET       ;DOUBLE CHECK TO ELIMINATE NOISE PROBLEMS.
5624 030222  103016                       BCC    60$             ;EXIT IF DIAG_FAIL BIT STILL CLEAR.
5625 030224  010501                       MOV    R5,R1           ;PASS THE REMAINING TIME-OUT VALUE.
5626 030226  000762                       BR     2$              ;LOOP TO CHECK AGAIN.
5627                                ;+
5628                                ; ERROR REPORTS:
5629                                ;-
5630                                    ;REPORT DIAGNOSTIC FAIL BIT BAD.
5631 030230  012767  002115  153532  4$:   MOV    #1101.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5632 030236  012701  011740                MOV    #EM1205,R1      ;SELECT ERROR MESSAGE.
5633 030242                                ERROR                  ;REPORT ERROR.          >>>>> ERROR #1101 <<<<<
     030242  104460                                                                              TRAP   C$ERROR
5634 030244  000405                       BR     60$             ;EXIT THE TEST.
5635
5636 030246  012767  002116  153514  50$:  MOV    #1102.,ERRNBR   ;SET THE ERROR NUMBER FOR TSABRT RTN.
5637 030254  004767  172202                JSR    PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
5638
5639 030260                          60$:  SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     030260  012700  000340                                                                    MOV    #PRI07,R0
     030264  104441                                                                           TRAP   C$SPRI
5640 030266  005067  152060                CLR    CTRLCF          ;INDICATE THAT WE COMPLETED A TEST.
5641 030272                                ENDTST
     030272                                                                          L10040:
     030272  104401                                                                           TRAP   C$ETST
```

```
                                        .SBTTL  HARDWARE TEST            - SELFTS -
                                        ;++ ******************************************************************************
                                        ;*              - SELF-TEST TEST -
                                        ;*      THIS TEST VERIFIES THAT THE DUT'S SELF-TEST EXECUTES WITHIN THE
                                        ;*      TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS
                                        ;*      COMPLETION.
                                        ;*
                                        ;-- ******************************************************************************
030274                                          BGNTST
030274                                                                                  T12::
030274                                          SETPRI  @PRIO5          ;ALLOW LTC INTERRUPTS.
030274  012700  000340                                                                  MOV     @PRIO5,R0
030300  104441                                                                        TRAP    C$SPRI
        000014                                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
030302  012767  000014  152016                   MOV    @TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (12)
030310  012767  177777  152034                   MOV    @-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
030316  012767  000001  153442                   MOV    @1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
030324  012767  011523  153440                   MOV    @EM1201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
030332  012767  016466  153434                   MOV    @ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                        ;*
                                        ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                        ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                        ;-
030340  012701  005670                           MOV    @3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
030344  012702  000040                           MOV    @BIT05,R2       ;WAITING FOR MASTER RESET BIT.
030350  005003                                   CLR    R3              ;WAITING FOR BIT TO CLEAR.
030352  016704  151666                           MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
030356  004767  167754                          JSR    PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030362  103062                                   BCC    50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
                                        ;*
                                        ; DETERMINE IF THE SELF-TEST TAKES TOO SHORT OR TOO LONG A TIME TO COMPLETE.
                                        ;  SET-UP A TIME-OUT OF 3 SECOND, IF MR IS CLEAR IN LESS THAN 1/2 SECOND, OR
                                        ;  GREATER THAN 3 SECONDS, REPORT THE ERROR.
                                        ;-
030364  012701  005670                           MOV    @3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
030370  010214                                   MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
030372  004767  167740                          JSR    PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030376  103030                                  BCC    4$              ;GO REPORT ERROR SELFTEST TOOK TOO LONG.
030400  012702  005670                           MOV    @3000.,R2
030404  160102                                   SUB    R1,R2           ;CALCULATE # OF MS SELFTEST TO COMPLETE.
030406  020227  000062                           CMP    R2,@50.
030412  002431                                  BLT    6$              ;SELFTEST SKIPPED?  YES, GO REPORT ERROR.
030414  020227  000764                           CMP    R2,@500.
030420  002434                                  BLT    8$              ;GO REP ERR IF SELFTEST COMPLETED IN < 1/2 SEC.
                                        ;*
                                        ; SELF-TEST COMPLETED WITHIN 1SEC TO 3 SECONDS.
                                        ;  CHECK THE STATE OF THE DIAGNOSTIC FAIL BIT, REPORT ERROR IF IT IS SET.
                                        ;-
030422  032714  020000                           BIT    @BIT13,(R4)     ;DETERMINE IF THE DIAG_FAIL BIT IS CLEAR.
030426  001406                                  BEQ    2$              ;SKIP ERROR REPORT IF BIT IS CLEAR.
                                           ;REPORT DIAGNOSTIC FAIL BIT BAD.
030430  012767  002264  153332                   MOV    @1204.,ERRNBR   ;SET ERROR NUMBER TO IN ERROR TABLE.
030436  012701  011740                           MOV    @EM1205,R1      ;SELECT THE ERROR MESSAGE.
030442                                          ERROR                  ;               >>>>> ERROR @1204 <<<<<
030442  104460                                                                        TRAP    C$ERROR
                                        ;*
                                        ; VERIFY THAT THE SELF-TEST CODES IN THE FIFO ARE "GOOD" CODES ,IE THE DUT
```

```
                                        ; SUCCESSFULLY COMPLETED THE SELF-TEST.
                                        ; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 1205 THRU 1209 <<<<<.
                                        ;-
030444  012767  002265  153316  2$:     MOV     #1205.,ERRNBR   ;SET ERROR NUMBER TO 1205.
030452  004767  171142                  JSR     PC,RSTRPT       ;CHECK SELF-TEST CODES IN THE FIFO.
030456  000431                          BR      60$             ;EXIT TEST.
                                        ;*
                                        ; ERROR REPORTS:
                                        ;-
                                    ;REPORT SELF-TEST TOOK TOO LONG TO COMPLETE.
030460  012767  002261  153302  4$:     MOV     #1201.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
030466  012701  011542                  MOV     #EM1202,R1      ;SELECT ERROR MESSAGE.
030472                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #1201 <<<<<
030472  104460                                                                                  TRAP    C$ERROR
030474  000422                          BR      60$             ;EXIT THE TEST.

                                    ;REPORT SELF-TEST DID NOT EXECUTE AFTER DUT RESET.
030476  012767  002262  153264  6$:     MOV     #1202.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
030504  012701  011704                  MOV     #EM1204,R1      ;SELECT ERROR MESSAGE.
030510                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #1202 <<<<<
030510  104460                                                                                  TRAP    C$ERROR

                                    ;REPORT SELF-TEST COMPETED TOO SOON.
030512  012767  002263  153250  8$:     MOV     #1203.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
030520  012701  011626                  MOV     #EM1203,R1      ;SELECT ERROR MESSAGE.
030524                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #1203 <<<<<
030524  104460                                                                                  TRAP    C$ERROR
030526  000405                          BR      60$             ;EXIT THE TEST.

030530  012767  002272  153232  50$:     MOV     #1210.,ERRNBR   ;SET THE ERROR NUMBER FOR TSABRT RTN.
030536  004767  171720                  JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.

030542                          60$:     SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
030542  012700  000340                                                                  MOV     #PRIO7,R0
030546  104441                                                                          TRAP    C$SPRI
030550  005067  151576                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
030554                                  ENDTST
030554                                                                          L10041:
030554  104401                                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST            - STFAIL -
                                ;** ************************************************************************
                                ;*              - SELF-TEST FAIL TEST -
                                ;*      THIS TEST VERIFIES THAT THE DUT WILL REPORT SELFTEST ERRORS VIA THE
                                ;*      FIFO.  AND THAT THE DIAGNOSTIC FAIL BIT WILL INDICATE THE ERROR.
                                ;*      THIS IS ACCOMPLISHED VIA A SOFTWARE 'HOOK' IN THE SELF-TEST, WHICH
                                ;*      FORCES A "PROC1 TO RAM ERROR" TO BE PLACED IN THE FIFO.
                                ;*
                                ;-- ************************************************************************
030556                                  BGNTST
030556                                                                          T13::
030556                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
030556  012700  000240                                                        MOV     #PRI05,R0
030562  104441                                                                TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
030564  012767  000015  151534           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (13)
030572  012767  177777  151552           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
030600  012767  000001  153160           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
030606  012767  011764  153156           MOV     #EM1301,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
030614  012767  016466  153152           MOV     #ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
030622  012767  002425  153140           MOV     #1301.,ERRNBR  ;SET ERROR NUMBER TO 1301.
                                ;*
                                ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                ;-
030630  012701  005670                   MOV     #3000.,R1      ;TIME-OUT VALUE IS 3.0 SECONDS.
030634  012702  000040                   MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
030640  005003                           CLR     R3             ;WAITING FOR BIT TO CLEAR.
030642  016704  151376                   MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
030646  004767  167464                   JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030652  103120                           BCC     50$            ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;*
                                ; RESET THE DUT, CHECK FOR ROM VERSION 0.
                                ;  IF VERSION 0 IS FOUND THEN EXIT THIS TEST.
                                ;-
030654  010214                           MOV     R2,(R4)        ;RESET THE DUT.
030656  004767  171452                   JSR     PC,SKPSTS      ;SKIP THE SELFTEST.
030662  012701  000764                   MOV     #500.,R1       ;PASS TIME-OUT VALUE OF 500 MILLISECS.
030666  004767  167444                   JSR     PC,MSLGET      ;WAIT FOR MR BIT TO CLEAR.
030672  103110                           BCC     50$            ;GO REPORT ERROR IF TIME-OUT OCCURRED.
030674  005267  153070                   INC     ERRNBR         ;SET ERROR NUMBER TO 1302.
030700  012700  000010                   MOV     #8.,R0         ;SET MAXIMUM READ COUNT.
030704  005003                           CLR     R3             ;CLEAR THE ROM VERSION 0 FLAG.
030706  016704  151334                   MOV     RBUFA,R4       ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
030712  011402                   2$:     MOV     (R4),R2        ;READ A CODE FROM THE FIFO.
030714  100077                           BPL     50$            ;GO REPORT ERROR IF THE FIFO IS EMPTY.
030716  042702  007402                   BIC     #7402,R2       ;REMOVE THE LINE NUMBER AND PROC INDICATOR.
030722  020227  170001                   CMP     R2,#170001     ;COMPARE WITH ROM VERSION #0 CODE.
030726  001002                           BNE     4$             ;ROM VERSION #0?  NO, SKIP SETTING FLAG.
030730  012703  177777                   MOV     #-1,R3         ;YES, SET THE ROM VERSION #0 FLAGS.
030734  005300                   4$:     DEC     R0             ;DECREMENT MAX READ COUNTER.
030736  001365                           BNE     2$             ;LOOP IF 8 CODES HAVE NOT BEEN READ.
030740  005703                           TST     R3             ;CHECK THE ROM VERSION #1 INDICATOR.
030742  100466                           BMI     60$            ;ROM VERSION 0 IN EITHER PROCESSOR?  YES, EXIT.
                                ;*
```

```
                                        ; RESET THE DUT, DELAY FOR 25 MILLI-SECONDS BEFORE WRITING THE FAIL_SELF_TEST
                                        ; CODE TO TBUFFCT REGISTER ON CHANNEL 0.
                                        ;-
030744  012777  000040  151272            MOV     @BIT05,@CSRA    ;SET DUT MASTER RESET BIT, SELECT CHANNEL 0.
030752  012704  000031                    MOV     @25.,R4         ;PASS DELAY PERIOD OF 25 MILLI SECS.
030756  004767  167314                    JSR     PC,DELAY        ;WAIT FOR SELFTEST TO INITIALISE.
030762  012777  146314  151272            MOV     @146314,@TXBFCA ;WRITE THE FAIL SELF-TEST CODE TO TBUFFCT REG.
                                        ;*
                                        ; WAIT UP TO 2 SECONDS FOR THE SELF-TEST TO COMPLETE.
                                        ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                        ;-
030770  005267  152774                    INC     ERRNBR          ;SET ERROR NUMBER TO 1303.
030774  012701  003720                    MOV     @2000.,R1       ;TIME-OUT VALUE IS 2.0 SECONDS.
031000  012702  000040                    MOV     @BIT05,R2       ;PASS THE BIT MAP OF THE BIT TO TEST.
031004  005003                            CLR     R3              ;SET UP THE EXPECTED STATE.
031006  016704  151232                    MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
031012  004767  167320                    JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
031016  103036                            BCC     50$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                        ;*
                                        ; VERIFY THE DIAGNOSTIC FAIL BIT IS SET, INDICATING THE ERROR.
                                        ; REPORT ERROR IF DIAGNOSTIC FAIL BIT IS CLEAR.
                                        ;-
031020  005267  152744                    INC     ERRNBR          ;SET ERROR NUMBER TO 1304.
031024  032714  020000                    BIT     @BIT13,(R4)     ;CHECK THE STATE OF THE DIAG_FAIL BIT.
031030  001425                            BEQ     10$             ;GO REPORT ERROR IF DIAG_FAIL BIT CLEAR.
                                        ;*
                                        ; REMOVE THE 8 SELF-TEST CODES FORM THE FIFO, AND VERIFY THAT AT LEAST
                                        ; ONE IS A PROC1 TO RAM ERROR CODE (231).
                                        ;-
031032  005267  152732                    INC     ERRNBR          ;SET ERROR NUMBER TO 1305.
031036  012700  000010                    MOV     @8.,R0          ;SET MAXIMUM READ COUNT.
031042  005001                            CLR     R1              ;CLEAR THE CORRECT CODE COUNTER.
031044  016704  151176                    MOV     RBUFA,R4        ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
031050  011402                6$:         MOV     (R4),R2         ;READ A CODE FROM THE FIFO.
031052  100020                            BPL     50$             ;GO REPORT ERROR IF THE FIFO IS EMPTY.
031054  042702  007400                    BIC     @7400,R2        ;REMOVE THE LINE NUMBER FROM THE CODE.
031060  120227  170231                    CMPB    R2,@170231      ;IS IT THE CORRECT ERROR CODE?.
031064  001001                            BNE     8$              ;SKIP NEXT INSTRUCTION, IF NOT A 231 CODE.
031066  005201                            INC     R1              ;INCREMENT COUNTER.
031070  005300                8$:         DEC     R0              ;DECREMENT MAX READ COUNTER.
031072  001366                            BNE     6$              ;LOOP IF 8 CODES HAVE NOT BEEN READ.
031074  005701                            TST     R1              ;WERE ANY 231 CODES FOUND?.
031076  001010                            BNE     60$             ;YES, THEN EXIT.
031100  005267  152664                    INC     ERRNBR          ;NO, SET ERROR NUMBER TO 1306 AND REPORT ERROR.
                                    ;REPORT SELF-TEST ERROR REPORTING BAD.
031104  012701  012010            10$:    MOV     @EM1302,R1      ;SELECT ERROR MESSAGE.
031110                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR <<<<<
031110  104460                                                                                    TRAP    C$ERROR
031112  000402                            BR      60$             ;EXIT THE TEST.
                                
031114  004767  171342            50$:    JSR     PC,TSABRT       ;REPORT NON-RELATED TEST ERROR.
                                
031120                            60$:    SETPRI  @PRI07          ;DISABLE ALL INTERRUPTS.
031120  012700  000340                                                                        MOV     @PRI07,R0
031124  104441                                                                                TRAP    C$SPRI
031126  005067  151220                    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
031132                                    ENDTST
```

HARDWARE TEST - STFAIL -

```
        031132
        031132  104401                                  L10042:
                                                                TRAP    C$FTST
```

```
                                .SBTTL  HARDWARE TEST           - ROMVER -
                                ;++ ************************************************************************
                                ;*              - ROM VERSION TEST -
                                ;*      THIS TEST VERIFIES THAT THE DUT'S SELF-TEST PLACES VALID ROM VERSION
                                ;*      NUMBERS IN THE FIFO AFTER IT HAS BEEN SKIPPED.  THE ROM VERSION NUMBERS
                                ;*      WILL BE REPORTED (ON THE FIRST PASS ONLY), IF AN AFFIRMATIVE ANSWER
                                ;*      WAS GIVEN TO THE SOFTWARE P-TABLE QUESTION.
                                ;*
                                ;-- ************************************************************************
031134                                  BGNTST
031134                                                                          T14::
031134                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
031134  012700  000240                                                        MOV     #PRIO5,R0
031140  104441                                                                TRAP    C$SPRI
        000016                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031142  012767  000016  151156          MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (14)
031150  012767  177777  151174          MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
031156  012767  000001  152602          MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
031164  012767  012047  152600          MOV     #EM1401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
031172  012767  016466  152574          MOV     #ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;*
                                ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                ;-
031200  012701  005670                  MOV     #3000.,R1      ;TIME-OUT VALUE IS 3.0 SECONDS.
031204  012702  000040                  MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
031210  005003                          CLR     R3             ;WAITING FOR BIT TO CLEAR.
031212  016704  151026                  MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
031216  004767  167114                  JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
031222  103131                          BCC     50$            ;ABORT THE TEST IF MR DID NOT CLEAR.
                                ;*
                                ; SET THE MASTER RESET BIT, AND SKIP THE SELF TEST.
                                ;-
031224  010214                          MOV     R2,(R4)        ;SET THE MASTER RESET BIT.
031226  004767  171102                  JSR     PC,SKPSTS      ;SKIP THE SELF TEST.
031232  012701  005670                  MOV     #3000.,R1      ;TIME-OUT VALUE IS 3.0 SECONDS.
031236  004767  167074                  JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
031242  103121                          BCC     50$            ;ABORT THE TEST IF MR DID NOT CLEAR.
                                ;*
                                ; REMOVE CHARACTERS FROM THE FIFO UNTIL EITHER:
                                ; (A) THE FIFO IS PURGED, GO REPORT THE ERROR.
                                ; (B) THE MAXIMUM TRY COUNTER IS ZERO, GO REPORT THE ERROR.
                                ; (C) PROC_1'S ROM VERSION NUMBER WAS FOUND BEFORE PROC_2'S, GO REPORT ERROR.
                                ; (D) BOTH ROM VERSION NUMBERS HAVE BEEN FOUND.
                                ;-
031244  012705  000040                  MOV     #4*NUMLNS,R5   ;SET MAXIMUM TRY COUNTER.
031250  012703  000143                  MOV     #99.,R3        ;SET AN INVALID ROM VERSION NUMBER FOR PROC_1.
031254  010304                          MOV     R3,R4          ;SET AN INVALID ROM VERSION NUMBER FOR PROC_2.
031256  012767  002571  152504          MOV     #1401.,ERRNBR  ;SET THE ERROR NUMBER TO 1401.
031264  012701  012077                  MOV     #EM1402,R1     ;SELECT MESSAGE TO BE REPORTED IF FIFO EMPTY.

031270  017702  150752          2$:     MOV     @RBUFA,R2      ;READ THE NEXT CHAR FROM THE FIFO.
031274  100077                          BPL     12$            ;GO REPORT ERROR IF FIFO EMPTY.
                                  ;*
                                  ; CHECK IF THE READ DATA IS A BMP CODE.
```

```
5892                                    ;-
5893 031276  012700  000301                     MOV     #301,R0         ;SET-UP A BIT MASK OF A BMP CODE.
5894 031302  040200                             BIC     R2,R0           ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
5895 031304  001003                            BNE     4$              ;BRANCH IF NOT A BMP CODE.
5896 031306  004767  170754                    JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
5897 031312  000435                            BR      8$              ;
5898                                    ;+
5899                                    ; CHECK IF THE READ DATA IS A SELF-TEST CODE.
5900                                    ;-
5901 031314  012700  000201            4$:      MOV     #201,R0         ;SET-UP A BIT MASK OF A SELFTEST CODE.
5902 031320  040200                             BIC     R2,R0           ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
5903 031322  001431                            BEQ     8$              ;BRANCH IF IT IS A SELFTEST CODE.
5904
5905                                    ;+
5906                                    ; THE READ DATA IS A ROM VERSION NUMBER, DETERMINE WHICH ONE IT IS.
5907                                    ;-
5908 031324  032702  000002                     BIT     #BIT1,R2        ;CHECK THE PROCESSOR NUMBER BIT IN THE CODE.
5909 031330  001407                            BEQ     6$              ;BRANCH IF IT IS PROC_1 ROM VERSION NUMBER.
5910 031332  010204                             MOV     R2,R4           ;SAVE PROC_2 ROM VERSION NUMBER.
5911 031334  042704  177603                     BIC     #177603,R4      ;CLEAR ANY UNWANTED BITS.
5912 031340  000241                             CLC                     ;CLEAR THE CARRY BIT.
5913 031342  006004                             ROR     R4              ;SHIFT THE CODES ALONG TO GET THE ROM
5914 031344  006004                             ROR     R4              ; VERSION NUMBER IN THE LOW 5 BITS.
5915 031346  000417                            BR      8$              ;
5916 031350  010203                    6$:      MOV     R2,R3           ;SAVE PROC_1 ROM VERSION NUMBER.
5917 031352  042703  177603                     BIC     #177603,R3      ;CLEAR ANY UNWANTED BITS.
5918 031356  000241                             CLC                     ;CLEAR THE CARRY BIT.
5919 031360  006003                             ROR     R3              ;SHIFT THE CODE ALONG TO GET THE ROM
5920 031362  006003                             ROR     R3              ; VERSION NUMBER IN THE LOW 5 BITS.
5921 031364  020427  000143                     CMP     R4,#99.         ;CHECK IF WE HAVE RECEIVE PROC_2 ROM CODE.
5922 031370  001016                            BNE     10$             ;GO REPORT BOTH ROM VERSION NUMBERS.
5923                                    ;+
5924                                    ; RECEIVED ROM VERSION NUMBERS OUT OF SEQUENCE.
5925                                    ;  IE, PROC_1'S ROM VERSION NUMBER FOUND IN THE FIFO BEFORE PROC_2'S.
5926                                    ;-
5927 031372  012701  012165                     MOV     #EM1403,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5928 031376  012767  002572  152364             MOV     #1402.,ERRNBR   ;SET THE ERROR NUMBER.
5929 031404  000433                            BR      12$             ;GO REPORT ERROR.
5930
5931 031406  005305                    8$:      DEC     R5              ;DECREMENT THE MAX TRY COUNTER.
5932 031410  001327                            BNE     2$              ;LOOP TO GET THE NEXT CHAR FROM THE FIFO.
5933 031412  012701  012240                     MOV     #EM1404,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5934 031416  012767  002573  152344             MOV     #1403.,ERRNBR   ;SET THE ERROR NUMBER.
5935 031424  000423                             BR      12$             ;GIVE UP, GO REPORT ERROR.
5936                                    ;+
5937                                    ; IF THIS IS THE FIRST PASS, AND SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
5938                                    ; THEN REPORT THE ROM VERSION NUMBERS TO THE OPERATOR.
5939                                    ;-
5940 031426  032767  000001  150572    10$:     BIT     #BIT0,OPTION    ;CHECK ON THE STATE OF THE SOFTWARE SWITCH.
5941 031434  001431                            BEQ     60$             ;EXIT IF NO ROM VERSION PRINTOUT WAS REQUESTED.
5942 031436  026727  150670  000001             CMP     PASCNT,#1       ;CHECK IF THIS IS THE FIRST PASS.
5943 031444  003025                            BGT     60$             ;EXIT IF ROM VERS HAVE ALREADY BEEN REPORTED.
5944 031446                                     PRINTB #EF1401,R3,R4    ;PRINT THE ROM VERSION NUMBERS.
     031446  010446                                                                    MOV     R4,-(SP)
     031450  010346                                                                    MOV     R3,-(SP)
     031452  012746  004212                                                            MOV     #EF1401,-(SP)
     031456  012746  000003                                                            MOV     #3,-(SP)
```

```
        031462  010600                                                          MOV     SP,R0
        031464  104414                                                          TRAP    C$PNTB
        031466  062706  000010                                                  ADD     #10,SP
5945    031472  000412                  BR      60$             ;EXIT THIS TEST.
5946                            ;
5947                            ; ERROR REPORTS:
5948                            ;
5949    031474  012767  016560  152272  12$:    MOV     #ER1401,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
5950    031502                          ERROR                           ;REPORT ERROR.          ***** ERROR *****
        031502  104460                                                          TRAP    C$ERROR
5951    031504  000405                  BR      60$
5952
5953    031506  012767  002575  152254  50$:    MOV     #1405.,ERRNBR   ;SET UP ERROR NUMBER FOR TSABRT RTN.
5954    031514  004767  170742                  JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
5955
5956    031520                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
        031520  012700  000340                                                  MOV     #PRI07,R0
        031524  104441                                                          TRAP    C$SPRI
5957    031526  005067  150620                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5958    031532                          ENDTST
        031532                                                          L10043:
        031532  104401                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - REGWRW -
                                ;** ************************************************************************
                                ;*                  - DEVICE REGISTER WORD ACCESS READ AND WRITE TEST -
                                ;*
                                ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
                                ;* CORRECTLY USING WORD ACCESSES.
                                ;*
                                ;-- ************************************************************************

                                    BGNTST
                                                                                        T15::
                                     SETPRI #PRIOS              ;ALLOW THE LTC TO INTERRUPT.
031534                                                                                  MOV     #PRIOS,R0
031540  104441                                                                         TRAP    C$SPRI
                                     TNUM == TNUM + 1           ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031542  012767  000017  150556      MOV     #TNUM,TSTNUM       ;SET UP THE TEST NUMBER.          (16)
031550  012767  177777  150574      MOV     #-1,CTRLCF         ;INDICATE THAT WE ARE WITHIN A TEST.
031556  012767  000001  152202      MOV     #1,ERRTYP          ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
031564  012767  003101  152176      MOV     #1601.,ERRNBR      ;SET UP ERROR NUMBER IN THE ERROR TABLE.
031572  012767  012443  152172      MOV     #EM1604,ERRMSG     ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
031600  005067  150656              CLR     ERSMRF             ;CLEAR THE ERROR SUMMARY FLAGS.
031604  012700  002464              MOV     #ERCNTB,R0
031610  004767  166362              JSR     PC,CLR16W          ;CLEAR THE ERROR COUNTER TABLE.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1601 <<<<<.
                                ;-
031614  004767  167634              JSR     PC,RESETT          ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
031620  103402                      BCS     .+6                ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
031622  000167  000116              JMP     60$                ;YES,  EXIT THE TEST.
                                ;*
                                ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR
                                ;-
031626  005267  152136              INC     ERRNBR             ;SET THE ERROR REPORT NUMBER TO 1602.
031632  012702  000017              MOV     #17,R2             ;SET LOOP COUNT.
031636  016704  150402              MOV     CSRA,R4            ;GET CSR ADDRESS.
031642  010214                  2$: MOV     R2,(R4)            ;WRITE COUNT TO CSR.
031644  011401                      MOV     (R4),R1            ;READ BACK THE CONTENTS OF THE CSR
031646  042701  177760              BIC     #177760,R1         ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
031652  020102                      CMP     R1,R2              ;CHECK FOR CORRECT DATA WRITTEN/READ.
031654  001406                      BEQ     4$                 ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
031656  012767  016710  152110      MOV     #ER1601,ERRBLK     ;SELECT THE PROPER ERROR REPORT ROUTINE.
031664  005003                      CLR     R3                 ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
031666  005005                      CLR     R5                 ;CAUSE REPORT OF LINE 0.
031670                              ERROR                      ;  >>>>> ERROR # 1602 <<<<<
031670  104460                                                                      TRAP    C$ERROR
031672  005302                  4$: DEC     R2                 ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
031674  002362                      BGE     2$                 ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
                                ;*
                                ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
                                ;   ACTIVE LINES.  BEFORE WRITING EACH PATTERN, CLEAR ALL THE BITS.
                                ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1603 - 1605 <<<<<.
                                ;-
031676  005267  152066              INC     ERRNBR             ;SET THE ERROR NUMBER TO 1603.
031702  005003                      CLR     R3                 ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
```

HARDWARE TEST - REGWRW -

```
6013 031704  012704  000002              MOV    #2,R4           ;INDICATE R/W ACCESS, CLEAR FIRST.
6014 031710  004767  167302              JSR    PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6015                                ;+
6016                                ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
6017                                ;   ACTIVE LINES.  BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
6018                                ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1606 - 1608 <<<<<.
6019                                ;-
6020 031714  012767  003106  152046        MOV    #1606.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6021 031722  005003                       CLR    R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6022 031724  005404                       NEG    R4              ;INDICATE R/W ACCESS, SET FIRST.
6023 031726  004767  167264              JSR    PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6024                                ;+
6025                                ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6026                                ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1609 <<<<<
6027                                ;-
6028 031732  012767  003111  152030        MOV    #1609.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
6029 031740  004767  167462              JSR    PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
6030 031744  005067  150402        60$:    CLR    CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
6031 031750                               ENDTST
     031750                                                                  L10044:
     031750  104401                                                              TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST            - REGWRM -
                                ;++ ******************************************************************************
                                ;*      - DEVICE REGISTER WORD ACCESS READ/MODIFY/WRITE TEST -
                                ;*
                                ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE WRITTEN CORRECTLY
                                ;* USING WORD READ/MODIFY/WRITE ACCESSES.
                                ;*
                                ;-- ******************************************************************************

                                        BGNTST
                                                                                        T16::
                                         SETPRI #PRIO5          ;ALLOW THE LTC TO INTERRUPT.
031752  012700  000240                                                          MOV     #PRIO5,R0
031756  104441                                                                  TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031760  012767  000020  150340          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (17)
031766  012767  177777  150356          MOV     #-1,CIRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
031774  012767  000001  151764          MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
032002  012767  003245  151760          MOV     #1701.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
032010  012767  012511  151754          MOV     #EM1701,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
032016  005067  150440                  CLR     ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
032022  012700  002464                  MOV     #ERCNTB,R0
032026  004767  166144                  JSR     PC,CLR16W       ;CLEAR THE ERROR COUNTER TABLE.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1701 <<<<<.
                                ;-
032032  004767  167416                  JSR     PC,RESETT       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
032036  103402                  BCS     .+6             ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
032040  000167  000122                  JMP     60$             ;YES,  EXIT THE TEST.
                                ;*
                                ; VERIFY READ/MODIFY/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR
                                ;-
                                         INC     ERRNBR         ;SET THE ERROR REPORT NUMBER TO 1702.
032044  005267  151720                  MOV     #17,R2          ;SET LOOP COUNT.
032050  012702  000017                  MOV     CSRA,R4         ;GET CSR ADDRESS.
032054  016704  150164          2$:     BIC     #17,(R4)        ;CLEAR THE DUT CSR USING READ/MODIFY/WRITE.
032060  042714  000017                  BIS     R2,(R4)         ;WRITE COUNT TO CSR USING READ/MODIFY/WRITE.
032064  050214                  MOV     (R4),R1         ;READ BACK THE CONTENTS OF THE CSR
032066  011401                  BIC     #177760,R1      ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
032070  042701  177760                  CMP     R1,R2           ;CHECK FOR CORRECT DATA WRITTEN/READ.
032074  020102                  BEQ     4$              ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
032076  001406                  ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
                                         MOV     #ER1601,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
032100  012767  016710  151666          CLR     R3              ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
032106  005003                  CLR     R5              ;CAUSE REPORT OF LINE 0.
032110  005005                  ERROR                   ;  >>>>> ERROR # 1702 <<<<<
032112                                                                          TRAP    C$ERROR
032112  104460
032114  005302          4$:     DEC     R2              ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
032116  002360                  BGE     2$              ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
                                ;*
                                ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
                                ;   ACTIVE LINES USING R/M/W.  BEFORE WRITING EACH PATTERN, CLEAR ALL THE BITS.
                                ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1703 - 1705 <<<<<.
                                ;-
032120  005267  151644                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 1703.
```

HARDWARE TEST - REGWRM -

```
6086 032124  005003                        CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6087 032126  012704  000001                MOV     #1,R4           ;INDICATE R/M/W ACCESS, CLEAR FIRST.
6088 032132  004767  167060                JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6089                                ;+
6090                                ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
6091                                ;   ACTIVE LINES USING R/M/W.  BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
6092                                ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1706 - 1708 <<<<<.
6093                                ;-
6094 032136  012767  003252  151624        MOV     #1706.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6095 032144  005003                        CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6096 032146  005404                        NEG     R4              ;INDICATE R/M/W ACCESS, SET FIRST.
6097 032150  004767  167042                JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6098                                ;+
6099                                ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6100                                ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1709 <<<<<
6101                                ;-
6102 032154  012767  003255  151606        MOV     #1709.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
6103 032162  004767  167240                JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
6104 032166  005067  150160          60$:  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
6105 032172                                ENDTST
     032172                                                                L10045:
     032172  104401                                                              TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - REGBRW -
                                ;++ ******************************************************************************
                                ;*              - DEVICE REGISTER BYTE ACCESS READ AND WRITE TEST -
                                ;*
                                ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
                                ;* CORRECTLY USING BYTE ACCESSES.
                                ;*
                                ;-- ******************************************************************************

                                        BGNTST
                                                                                T17::
                                         SETPRI #PRIO5          ;ALLOW THE LTC TO INTERRUPT.
032174  012700  000240                                                          MOV     #PRIO5,R0
032200  104441                                                                  TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
000021
032202  012767  000021  150116          MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (18)
032210  012767  177777  150134          MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
032216  012767  000001  151542          MOV     #1,ERRTYP      ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
032224  012767  003411  151536          MOV     #1801.,ERRNBR  ;SET UP ERROR NUMBER IN THE ERROR TABLE.
032232  012767  012566  151532          MOV     #EM1801,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
032240  005067  150216                  CLR     ERSMRF         ;CLEAR THE ERROR SUMMARY FLAGS.
032244  012700  002464                  MOV     #ERCNTB,R0
032250  004767  165722                  JSR     PC,CLR16W      ;CLEAR THE ERROR COUNTER TABLE.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1801 <<<<<.
                                ;-
032254  004767  167174                  JSR     PC,RESETT      ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
032260  103402                          BCS     .+6            ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
032262  000167  000172                  JMP     60$            ;YES,  EXIT THE TEST.
032266  012767  003412  151474          MOV     #1802.,ERRNBR  ;SET THE ERROR REPORT NUMBER TO 1802.
                                ;*
                                ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR.
                                ; USE BYTE ACCESSES.
                                ;-
032274  012702  000017                  MOV     #17,R2         ;SET LOOP COUNT.
032300  016704  147740                  MOV     CSRA,R4        ;GET CSR ADDRESS.
032304  110214                  2$:     MOVB    R2,(R4)        ;WRITE COUNT TO CSR.
032306  111401                          MOVB    (R4),R1        ;READ BACK THE CONTENTS OF THE CSR
032310  042701  177760                  BIC     #177760,R1     ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
032314  020102                          CMP     R1,R2          ;CHECK FOR CORRECT DATA WRITTEN/READ.
032316  001406                          BEQ     4$             ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                    ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
032320  012767  016710  151446          MOV     #ER1601,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
032326  005003                          CLR     R3             ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
032330  005005                          CLR     R5             ;CAUSE REPORT OF LINE 0.
032332                                  ERROR                  ;  >>>>> ERROR # 1802 <<<<<
032332  104460                                                                  TRAP    C$ERROR
032334  005302                  4$:     DEC     R2             ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
032336  002362                          BGE     2$             ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
                                ;*
                                ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
                                ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
                                ;   EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
                                ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1803 - 1805 <<<<<.
                                ;-
```

```
                                        INC     ERRNBR          ;SET THE ERROR NUMBER TO 1803.
                                        MOV     #-1,R3          ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
                                        MOV     #2,R4           ;INDICATE R/W ACCESS, CLEAR FIRST.
                                        JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                ;*
                                ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
                                ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
                                ;   EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
                                ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1806 - 1808 <<<<<.
                                ;-
                                        MOV     #1806.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
                                        NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
                                        JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                ;*
                                ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
                                ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
                                ;   EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
                                ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1809 - 1811 <<<<<.
                                ;-
                                        MOV     #1809.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
                                        NEG     R3              ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
                                        NEG     R4              ;INDICATE R/W ACCESS, SET FIRST.
                                        JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                ;*
                                ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
                                ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
                                ;   EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
                                ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1812 - 1814 <<<<<.
                                ;-
                                        MOV     #1812.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
                                        NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
                                        JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                ;*
                                ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
                                ;    THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1815 <<<<<
                                ;-
                                        MOV     #1815.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
                                        JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
                                60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
                                        ENDTST
                                                                                        L10046:
                                                                                                TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - REGBRM -
                                ;** ************************************************************************
                                ;*              - DEVICE REGISTER BYTE ACCESS READ/MODIFY/WRITE TEST -
                                ;*
                                ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
                                ;* CORRECTLY USING BYTE ACCESSES IN READ/MODIFY/WRITE MODE.
                                ;*
                                ;-- ************************************************************************

                                    BGNTST
                                                                                        T18::
                                     SETPRI #PRI05          ;ALLOW THE LTC TO INTERRUPT.
032446  012700  000240                                                          MOV     #PRI05,R0
032452  104441                                                                  TRAP    C$SPRI
                                     TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
032454  012767  000022  147644      MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (19)
032462  012767  177777  147662      MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
032470  012767  000001  151270      MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
032476  012767  003555  151264      MOV     #1901.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
032504  012767  012634  151260      MOV     #EM1901,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
032512  005067  147744              CLR     ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
032516  012700  002464              MOV     #ERCNTB,R0
032522  004767  165450              JSR     PC,CLR16W       ;CLEAR THE ERROR COUNTER TABLE.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1901 <<<<<.
                                ;-
032526  004767  166722              JSR     PC,RESETT       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
032532  103402                      BCS     .+6             ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
032534  000167  000156              JMP     60$             ;YES,  EXIT THE TEST.
032540  012767  003556  151222      MOV     #1902.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 1902.
                                ;*
                                ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR.
                                ; USE BYTE ACCESSES.
                                ;-
032546  012702  000017              MOV     #17,R2          ;SET LOOP COUNT.
032552  016704  147466              MOV     CSRA,R4         ;GET CSR ADDRESS.
032556  142714  000017      2$:     BICB    #17,(R4)        ;CLEAR THE DUT CSR USING READ/MODIFY/WRITE.
032562  150214                      BISB    R2,(R4)         ;WRITE COUNT TO CSR USING READ/MODIFY/WRITE.
032564  111401                      MOVB    (R4),R1         ;READ BACK THE CONTENTS OF THE CSR
032566  042701  177760              BIC     #177760,R1      ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
032572  020102                      CMP     R1,R2           ;CHECK FOR CORRECT DATA WRITTEN/READ.
032574  001406                      BEQ     4$              ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
032576  012767  016710  151170      MOV     #ER1601,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
032604  005003                      CLR     R3              ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
032606  005005                      CLR     R5              ;CAUSE REPORT OF LINE 0.
032610                              ERROR                   ;  >>>>> ERROR # 1902 <<<<<
032610  104460                                                                  TRAP    C$ERROR
032612  005302              4$:     DEC     R2              ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
032614  002360                      BGE     2$              ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
                                ;*
                                ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
                                ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/MODIFY/WRITE ACCESSES.  BEFORE
                                ;   WRITING EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
                                ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1903 - 1905 <<<<<.
```

```
                                        ;-
032616  005267  151146                      INC     ERRNBR          ;SET THE ERROR NUMBER TO 1903.
032622  012703  177777                      MOV     #-1,R3          ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
032626  012704  000001                      MOV     #1,R4           ;INDICATE R/M/W ACCESS, CLEAR FIRST.
032632  004767  166360                    JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                        ;*
                                        ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
                                        ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/MODIFY/WRITE ACCESSES.  BEFORE
                                        ;   WRITING EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
                                        ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1906 - 1908 <<<<<.
                                        ;-
032636  012767  003562  151124              MOV     #1906.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
032644  005403                            NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
032646  004767  166344                    JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                        ;*
                                        ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
                                        ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/MODIFY/WRITE ACCESSES.  BEFORE
                                        ;   WRITING EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
                                        ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1909 - 1911 <<<<<.
                                        ;-
032652  012767  003565  151110              MOV     #1909.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
032660  005403                            NEG     R3              ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
032662  005404                            NEG     R4              ;INDICATE R/M/W ACCESS, SET FIRST.
032664  004767  166326                    JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                        ;*
                                        ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
                                        ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/MODIFY/WRITE ACCESSES.  BEFORE
                                        ;   WRITING EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
                                        ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1912 - 1914 <<<<<.
                                        ;-
032670  012767  003570  151072              MOV     #1912.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
032676  005403                            NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
032700  004767  166312                    JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                        ;*
                                        ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
                                        ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1915 <<<<<
                                        ;-
032704  012767  003573  151056              MOV     #1915.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
032712  004767  166510                    JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
032716  005067  147430            60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
032722                                    ENDTST
032722                                                                    L10047:
032722  104401                                                                    TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE TEST            - IDBIT -
                                        ;++ ************************************************************************
                                        ;*              - DEVICE REGISTER ID BIT TEST -
                                        ;*
                                        ;*      THIS TEST VERIFIES THAT THE DUT STAT REGISTER ID BIT READS AS CLEAR.
                                        ;*
                                        ;-- ************************************************************************

                                                BGNTST
                                                                                                        T19::
                                                 SETPRI @PRIO5          ;ALLOW THE LTC TO INTERRUPT.
032724  012700  000240                                                                          MOV     @PRIO5,R0
032730  104441                                                                                  TRAP    C$SPRI
                000023                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
032732  012767  000023  147366                  MOV     @TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (20)
032740  012767  177777  147404                  MOV     @-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
032746  012767  000001  151012                  MOV     @1,ERRTYP      ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
032754  012767  003721  151006                  MOV     @2001.,ERRNBR  ;SET UP ERROR NUMBER IN THE ERROR TABLE.
032762  012767  012711  151002                  MOV     @EM2001,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
                                        ;*
                                        ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                        ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                        ; THIS SUBROUTINE REPORTS ERRORS >>>>> 2001 <<<<<.
                                        ;-
032770  004767  166460                          JSR     PC,RESETT      ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
032774  103016                                BCC     60$            ;FATAL RESET ERROR?  YES,  EXIT THE TEST.
                                        ;*
                                        ; READ THE STAT REGISTER ID BIT AND VERIFY THAT IT IS CLEAR.
                                        ;-
032776  017701  147250                          MOV     @STATA,R1      ;READ THE STAT REGISTER CONTENTS.
033002  032701  000400                          BIT     #BIT8,R1       ;CHECK THE ID BIT.
033006  001411                                BEQ     60$            ;ID BIT CLEAR?  YES, EXIT THE TEST.
033010  012767  003722  150752                  MOV     #2002.,ERRNBR  ;NO,  SET THE ERROR REPORT NUMBER TO 2002.
033016  012701  012753                          MOV     #EM2002,R1     ;GET THE PROPER ERROR MESSAGE.
033022  012767  016466  150744                  MOV     #ER0503,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
033030                                          ERROR                  ;ERROR NUMBER >>>>> 2002 <<<<<
033030  104460                                                                                  TRAP    C$ERROR
033032  005067  147314                60$:      CLR     CTRLCF         ;INDICATE THAT WE COMPLETED THE TEST.
033036                                          ENDTST
                                                                                        L10050:
033036  104401                                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - NOTXDV -
                                ;++ ************************************************************************
                                ;*                      - NO TX_DATA_VALID/NO TX_ACTION TEST -
                                ;*      THIS TEST VERIFIES THAT IF A DATA WORD IS WRITTEN WITHOUT THE
                                ;*      TX_DATA_VALID BIT SET, NO TX_ACTION WILL BE GENERATED.
                                ;*      TO ENSURE DATA IS NOT ACCIDENTALLY TRANSMITTED, THE TEST IS PERFORMED
                                ;*      IN INTERNAL LOOPBACK, AND ON ALL ACTIVE LINES.
                                ;*
                                ;-- ************************************************************************
                                        BGNTST
033040                                                                          T20::
                                        SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
033040  012700  000240                                                          MOV     #PRIO5,R0
033044  104441                                                                  TRAP    C$SPRI
                000024                  TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
033046  012767  000024  147252          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (21)
033054  012767  177777  147270          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
033062  012767  000001  150676          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
033070  012767  004065  150672          MOV     #2101.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
033076  012767  013026  150666          MOV     #EM2101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
033104  012767  017420  150662          MOV     #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 2101 <<<<<.
                                ;-
033112  004767  165036                  JSR     PC,CLNRST       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
033116  103054                          BCC     60$             ;RESET FAILURE?, ABORT THIS TEST.
033120  005267  150644                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 2102.
                                ;*
                                ; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
                                ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
                                ; 2 STOP BITS.
                                ; DISABLE TRANSMITTERS ON ALL ACTIVE LINES.
                                ;-
033124  016705  147106                  MOV     ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
033130  012700  000200                  MOV     #200,R0         ;PASS THE LNCTRL CONTENTS.
033134  004767  170370                  JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
033140  012700  177670                  MOV     #177670,R0      ;PASS THE LPR CONTENTS.
033144  004767  170434                  JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
033150  012704  000012                  MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECS.
033154  004767  165116                  JSR     PC,DELAY        ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
033160  004767  167410                  JSR     PC,TXDSBL       ;DISABLE TRANSMITTERS ON ALL ACTIVE LINES.
                                ;*
                                ; TEST ALL ACTIVE LINES INDIVIDUALLY.
                                ; WRITE A DATA WORD TO THE TXCHAR REGISTER WITH TX_DATA_VALID CLEAR.
                                ; VERIFY NO TX_ACTION IS GENERATED.
                                ;-
033164  016705  147046                  MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
033170  005004                          CLR     R4              ;CLEAR THE LINE NUMBER COUNTER.
033172  000241                  2$:     CLC                     ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
033174  006005                          ROR     R5              ;SHIFT THE BIT MAP INTO THE CARRY BIT.
033176  103020                          BCC     4$              ;DO NOT TEST THE LINE IF IT IS INACTIVE.
                                ;*
                                ; SELECT THE LINE UNDER TEST.
                                ;  WRITE DATA WORD (ASCII <LF>) TO TXCHR REGISTER WITH THE MOST SIGNIFICANT
                                ;  BIT (TX_DATA_VALID) CLEAR.
```

HARDWARE TEST - NOTXDV -

```
6386                                            ;-
6387 033200  010477  147040                       MOV    R4,@CSRA          ;SELECT THE LINE CURRENTLY UNDER TEST.
6388 033204  012777  000012  147034               MOV    #12,@TXCHA        ;WRITE THE DATA WORD TO THE DUT'S TXCHAR REG.
6389                                            ;+
6390                                            ; WAIT FOR A TX_ACTION TO BE RETURNED, REPORT ERROR IF TX_ACTION FOUND
6391                                            ; BEFORE TIME-OUT OCCURS.
6392                                            ;-
6393 033212  012701  170002                       MOV    #170002,R1        ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
6394 033216  016702  147022                       MOV    CSRA,R2           ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6395 033222  004767  170032                      JSR     PC,WAIBIS         ;WAIT FOR TX_ACTION TO COME BACK.
6396 033226  103004                              BCC     4$                ;SKIP ERROR REPORT IF TX-ACTION NOT FOUND.
6397
6398 033230  010401                              MOV     R4,R1             ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6399 033232  012702  013071                      MOV     #EM2102,R2        ;PASS THE ERROR MESSAGE TO BE REPORTED.
6400                                                                       ;"TX_ACT FOUND AFTER INVALID DATA WORD WRITTEN"
6401 033236                                      ERROR                     ;              >>>>> ERROR #4102 <<<<<.
     033236  104460                                                                                  TRAP    C$ERROR
6402                                    ;+
6403                                    ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6404                                    ;-
6405 033240  005204                     4$:      INC    R4                ;INCREMENT THE LINE NUMBER COUNTER.
6406 033242  005705                              TST    R5                ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
6407 033244  001352                             BNE     2$                ;YES; BRANCH TO TEST THE NEXT LINE.
6408 033246  000400                             BR      60$               ;NO; EXIT THIS TEST.
6409
6410 033250  005067  147076             60$:     CLR    CTRLCF            ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6411 033254                                      ENDTST
     033254                                                                                  L10051:
     033254  104401                                                                              TRAP    C$ETST
```

```
6413                                    .SBTTL  HARDWARE TEST           - TXDVAL -
6414                                    ;** ************************************************************************
6415                                    ;*                      - TX_DATA_VALID/TX_ACTION TEST -
6416                                    ;*      THIS TEST VERIFIES THAT IF A DATA WORD IS WRITTEN TO THE TXCHAR REGISTER
6417                                    ;*      WITH THE TX_DATA_VALID BIT SET, A CORRESPONDING TX_ACTION WILL BE
6418                                    ;*      GENERATED.
6419                                    ;*      TO ENSURE DATA IS NOT ACCIDENTALLY TRANSMITTED, THE TEST IS PERFORMED
6420                                    ;*      IN INTERNAL LOOPBACK, AND ON ALL ACTIVE LINES.
6421                                    ;*
6422                                    ;-- ************************************************************************
6423
6424 033256                                      BGNTST
        033256                                                                  T21::
6425 033256                                      SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
        033256  012700  000240                                                  MOV     #PRI05,R0
        033262  104441                                                          TRAP    C$SPRI
6426            000025                            TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6427 033264  012767  000025  147034               MOV     #TNUM,TSTNUM  ;SET UP THE TEST NUMBER.        (22)
6428 033272  012767  177777  147052               MOV     #-1,CTRLCF    ;INDICATE THAT WE ARE IN A TEST.
6429 033300  012767  000001  150460               MOV     #1,ERRTYP     ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6430 033306  012767  004231  150454               MOV     #2201.,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
6431 033314  012767  013165  150450               MOV     #EM2201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
6432 033322  012767  017420  150444               MOV     #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6433                                    ;*
6434                                    ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
6435                                    ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6436                                    ; THIS SUBROUTINE REPORTS ERROR >>>>> 2201 <<<<<.
6437                                    ;-
6438 033330  004767  164620                       JSR     PC,CLNRST     ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
6439 033334  103066                               BCC     60$           ;RESET FAILURE?, ABORT THIS TEST.
6440                                    ;*
6441                                    ; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
6442                                    ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
6443                                    ; 2 STOP BITS.
6444                                    ; DISABLE TRANSMITTERS ON ALL ACTIVE LINES.
6445                                    ;-
6446 033336  016705  146674                       MOV     ACTLNS,R5     ;PASS THE ACTIVE LINE BIT MAP.
6447 033342  012700  000200                       MOV     #200,R0       ;PASS THE LNCTRL CONTENTS.
6448 033346  004767  170156                       JSR     PC,WTWLNC     ;INITIALISE THE LNCTRL REGISTERS.
6449 033352  012700  177670                       MOV     #177670,R0    ;PASS THE LPR CONTENTS.
6450 033356  004767  170222                       JSR     PC,WTWLPR     ;INITIALSE THE LPR REGISTERS ON ALL LINES.
6451 033362  012704  000012                       MOV     #10.,R4       ;PASS DELAY TIME OF 10 MILLI-SECONDS.
6452 033366  004767  164704                       JSR     PC,DELAY      ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
6453 033372  004767  167176                       JSR     PC,TXDSBL     ;DISABLE TRANSMITTERS ON ALL ACTIVE LINES.
6454                                    ;*
6455                                    ; TEST ALL ACTIVE LINES INDIVIDUALLY.
6456                                    ; WRITE A DATA WORD TO THE TXCHAR REGISTER WITH TX_DATA_VALID SET.
6457                                    ; VERIFY THAT A CORRESPONDING TX_ACTION IS GENERATED.
6458                                    ;-
6459 033376  016705  146634                       MOV     ACTLNS,R5     ;GET THE ACTIVE LINE BIT MAP.
6460 033402  005004                               CLR     R4            ;CLEAR THE LINE NUMBER COUNTER.
6461 033404  012767  004232  150356  2$:          MOV     #2202.,ERRNBR ;SET THE ERROR NUMBER TO 2202.
6462 033412  000241                               CLC                   ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
6463 033414  006005                               ROR     R5            ;SHIFT THE BIT MAP INTO THE CARRY BIT.
6464 033416  103032                              BCC     8$            ;DO NOT TEST THE LINE IF IT IS INACTIVE.
6465                                        ;*
6466                                        ; SELECT THE LINE UNDER TEST.
```

```
                                        ;  WRITE DATA WORD (ASCII <LF>) TO TXCHR REGISTER WITH THE MOST SIGNIFICANT
                                        ;  BIT (TX_DATA_VALID) SET.
                                        ;-
033420  010477  146620                    MOV     R4,@CSRA          ;SELECT THE LINE CURRENTLY UNDER TEST.
033424  012777  100012  146614            MOV     #100012,@TXCHA    ;WRITE THE DATA WORD TO THE DUT'S TXCHAR REG.
                                        ;*
                                        ;  WAIT FOR A TX_ACTION TO BE RETURNED. REPORT ERROR IF NO TX_ACTION
                                        ;  FOUND BEFORE TIME-OUT OCCURS.
                                        ;-
033432  012701  170002                    MOV     #170002,R1        ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
033436  016702  146602                    MOV     CSRA,R2           ;PASS THE ADDRESS OF THE REGISTER TO TEST.
033442  004767  167612                    JSR     PC,WAIBIS         ;WAIT FOR TX_ACTION TO COME BACK.
033446  103403                            BCS     4$                ;SKIP ERROR REPORT IF TX-ACTION FOUND.
033450  012702  013222                    MOV     @EM2202,R2        ;PASS THE ERROR MESSAGE TO BE REPORTED.
                                                                  ;"NO TX_ACT FOUND AFTER VALID DATA WORD TX'D".
033454  000411                            BR      6$                ;GO REPORT THE ERROR.
                                        ;*
                                        ;  VERIFY TX_ACTION RETURNED FROM CORRECT LINE.
                                        ;-
033456  005267  150306          4$:       INC     ERRNBR            ;INCREMENT ERROR NUMBER TO 2103.
033462  000302                            SWAB    R2                ;GET THE LINE NUMBER IN THE LOW BYTE.
033464  042702  177760                    BIC     #177760,R2        ;CLEAR THE UNWANTED BITS.
033470  020204                            CMP     R2,R4             ;IS IT THE CORRECT LINE NUMBER?.
033472  001404                            BEQ     8$                ;YES; SKIP THE ERROR REPORT.
033474  012702  013314                    MOV     @EM2203,R2        ;PASS THE ERROR MESSAGE TO BE REPORTED.
                                                                  ;  "INCORRECT LINE # RETURNED WITH TX_ACT"
033500  010401                  6$:       MOV     R4,R1             ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
033502                                    ERROR                     ;                >>>>> ERROR <<<<<.
033502  104460                                                                              TRAP    C$ERROR
                                        ;*
                                        ;  VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
                                        ;-
033504  005204                  8$:       INC     R4                ;INCREMENT THE LINE NUMBER COUNTER.
033506  005705                            TST     R5                ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
033510  001335                            BNE     2$                ;YES; BRANCH TO TEST THE NEXT LINE.

033512  005067  146634          60$:      CLR     CTRLCF            ;INDICATE THAT WE ARE NOT WITHIN A TEST.
033516                                    ENDTST
033516                                                                                  L10052:
033516  104401                                                                                TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST                - TXENBI-
                                ;** ************************************************************************
                                ;*                          - TX_ENABLE (INACTIVE) TEST -
                                ;*      THIS TEST VERIFIES THAT WHEN THE LINE UNDER TEST'S TX_ENABLE BIT IS
                                ;*      CLEAR, TRANSMISSION WILL NOT TAKE PLACE ON THAT LINE.
                                ;*      THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, AND ON ALL ACTIVE LINES.
                                ;*
                                ;-- ************************************************************************

                                        BGNTST
                                                                                T22::
                                        SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
                                                                                MOV     #PRI05,R0
                                                                                TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
                                         MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (23)
                                         MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
                                         MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
                                         MOV    #2301.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
                                         MOV    #EM2301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
                                         MOV    #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 2301 <<<<<.
                                ;-
                                        JSR     PC,CLNRST       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
                                        BCC     60$             ;RESET FAILURE?, ABORT THIS TEST.
                                ;+
                                ; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
                                ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
                                ; 2 STOP BITS.
                                ; ENABLE TRANSMITTERS ON ALL LINES.
                                ;-
                                         MOV    ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
                                         MOV    #200,R0         ;PASS THE LNCTRL CONTENTS.
                                        JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
                                         MOV    #177670,R0      ;PASS THE LPR CONTENTS.
                                        JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
                                         MOV    #10.,R4         ;PASS DELAY TIME OF 10 MILLI-SECONDS.
                                        JSR     PC,DELAY        ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
                                         MOV    #MAPLNS,R5      ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
                                        JSR     PC,TXENBL       ;ENABLE TRANSMITTERS ON ALL LINES.
                                ;+
                                ; TEST ALL ACTIVE LINES INDIVIDUALLY.
                                ; DISABLE TRANSMISSION ON EACH ACTIVE LINE.
                                ;-
                                         MOV    #1,R3           ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
                                         CLR    R4              ;CLEAR THE LINE NUMBER COUNTER.
                                2$:      MOV    #2302.,ERRNBR   ;SET THE ERROR NUMBER TO 2302.
                                        BIT     R3,ACTLNS       ;CHECK IF THE LINE IS ACTIVE.
                                        BEQ     6$              ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
                                    ;+
                                    ; CLEAR THE TX_ENABLE BIT IN TBUFFAD2 REGISTER.
                                    ; SELECT THE LINE UNDER TEST.
                                    ;   VERIFY IT IS CLEAR, REPORT ERROR IF SET.
                                    ;-
```

```
6560 033666  010305                        MOV    R3,R5           ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6561 033670  004767  166700                JSR    PC,TXDSBL       ;DISABLE TRANSMISSION ON THE LINE UNDER TEST.
6562 033674  010477  146344                MOV    R4,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
6563 033700  005777  146354                TST    @TXAD2A         ;VERIFY THE TX_ENABLE BIT IS SET.
6564 033704  100433                        BMI    4$              ;GO REPORT ERROR IF TX_ENABLE BIT SET.
6565                                      ;+
6566                                      ; WRITE DATA WORD (ASCII <LF>) TO TXCHR REGISTER.
6567                                      ; WAIT FOR A TX_ACTION TO BE RETURNED, REPORT ERROR IF A TX_ACTION
6568                                      ; IS FOUND BEFORE TIME-OUT OCCURS.
6569                                      ;-
6570 033706  012767  004377  150054        MOV    #2303.,ERRNBR   ;SET ERROR NUMBER TO 2303.
6571 033714  012777  100012  146324        MOV    #100012,@TXCHA  ;WRITE THE DATA WORD TO THE DUT'S TXCHAR REG.
6572 033722  012701  170002                MOV    #170002,R1      ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
6573 033726  016702  146312                MOV    CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6574 033732  004767  167322                JSR    PC,WAIBIS       ;WAIT FOR TX_ACTION TO COME BACK.
6575 033736  103016                        BCC    4$              ;GO REPORT ERROR IF NO TX-ACTION FOUND.
6576                                      ;+
6577                                      ; WAIT FOR THE DATA TO APPEAR IN THE FIFO, REPORT ERROR IF DATA FOUND.
6578                                      ;-
6579 033740  005267  150024                INC    ERRNBR          ;SET ERROR NUMBER TO 2304.
6580 033744  012701  070012                MOV    #70012,R1       ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
6581 033750  016702  146270                MOV    CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6582 033754  004767  167300                JSR    PC,WAIBIS       ;WAIT FOR RX_DATA_AVAILABLE TO SET.
6583 033760  103405                        BCS    4$              ;REPORT ERROR IF DATA RECEIVED IN THE FIFO.
6584 033762  005267  150002                INC    ERRNBR          ;SET ERROR NUMBER TO 2305.
6585 033766  017702  146254                MOV    @RBUFA,R2       ;READ THE DATA FROM THE FIFO.
6586 033772  100004                        BPL    6$              ;SKIP ERROR REPORT IF DATA IS THERE.
6587
6588 033774  010401              4$:       MOV    R4,R1           ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6589 033776  012702  013457                MOV    #EM2302,R2      ;PASS THE MESSAGE TO BE REPORTED.
6590                                                              ;"TX_ENABLE BIT BAD ON LINE: NN".
6591 034002                                ERROR                  ;          >>>>> ERROR <<<<<.
     034002  104460                                                                TRAP    C$ERROR
6592                                      ;+
6593                                      ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6594                                      ;-
6595 034004  000241              6$:       CLC                    ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
6596 034006  006103                        ROL    R3              ;SHIFT THE BIT MAP FOR THE NEXT LINE.
6597 034010  005204                        INC    R4              ;INCREMENT THE LINE NUMBER COUNTER.
6598 034012  020427  000010                CMP    R4,#NUMLNS      ;HAVE ALL THE LINES BEEN TESTED?.
6599 034016  002715                        BLT    2$              ;NO; BRANCH TO TEST THE NEXT LINE.
6600
6601 034020  005067  146326        60$:    CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6602 034024                                ENDTST
     034024                                                                  L10053:
     034024  104401                                                                TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST            - TXENBA-
                                ;** ************************************************************************
                                ;*                          - TX_ENABLE (ACTIVE) TEST -
                                ;*      THIS TEST VERIFIES THAT WHEN THE TX_ENABLE BIT IS SET IN THE APPRORIATE
                                ;*      LINE REGISTER, TRANSMISSION WILL TAKE PLACE ON THAT LINE.
                                ;*      THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, AND ON ALL ACTIVE LINES.
                                ;*
                                ;-- ************************************************************************

                                        BGNTST
                                                                                        T23::
                                        SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
                                                                                        MOV     #PRI05,R0
                                                                                        TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
                                         MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (24)
                                         MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
                                         MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
                                         MOV    #2401.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
                                         MOV    #EM2401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
                                         MOV    #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 2401 <<<<<.
                                ;-
                                        JSR     PC,CLNRST       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
                                        BCC     60$             ;RESET FAILURE?, ABORT THIS TEST.
                                ;+
                                ; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
                                ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
                                ; 2 STOP BITS.
                                ; DISABLE TRANSMITTERS ON ALL LINES.
                                ;-
                                         MOV    ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
                                         MOV    #200,R0         ;PASS THE LNCTRL CONTENTS.
                                        JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
                                         MOV    #177670,R0      ;PASS THE LPR CONTENTS.
                                        JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
                                         MOV    #10.,R4         ;PASS DELAY TIME OF 10 MILLI-SECONDS.
                                        JSR     PC,DELAY        ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
                                         MOV    #MAPLNS,R5      ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
                                        JSR     PC,TXDSBL       ;DISABLE TRANSMITTERS ON ALL LINES.
                                ;+
                                ; TEST ALL ACTIVE LINES INDIVIDUALLY.
                                ; ENABLE TRANSMISSION ON EACH ACTIVE LINE.
                                ;-
                                         MOV    #1,R3           ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
                                         CLR    R4              ;CLEAR THE LINE NUMBER COUNTER.
                                2$:      MOV    #2402.,ERRNBR   ;SET THE ERROR NUMBER TO 2402.
                                         BIT    R3,ACTLNS       ;CHECK IF THE LINE IS ACTIVE.
                                        BEQ     8$              ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
                                    ;+
                                    ; SELECT THE LINE UNDER TEST.
                                    ;  SET THE TX_ENABLE BIT IN TBUFFAD2 REGISTER.
                                    ;   VERIFY IT IS SET, REPORT ERROR IF CLEAR.
                                    ;-
```

```
6658 034174  010305                        MOV    R3,R5           ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6659 034176  004767  166466                JSR    PC,TXENBL       ;ENABLE TRANSMISSION ON THE LINE UNDER TEST.
6660 034202  012705  000012                MOV    #10.,R5         ;SET TXCHAR/LOOP COUNT TO 10.
6661 034206  010477  146032                MOV    R4,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
6662 034212  005777  146042                TST    @TXAD2A         ;VERIFY THE TX_ENABLE BIT IS SET.
6663 034216  100045                        BPL    6$              ;GO REPORT ERROR IF TX_ENABLE BIT CLEAR.
6664                                      ;+
6665                                      ; WRITE DATA WORD (ASCII <LF>) TO TXCHR REGISTER.
6666                                      ; WAIT FOR A TX_ACTION TO BE RETURNED, REPORT ERROR IF NO TX_ACTION
6667                                      ; FOUND BEFORE TIME-OUT OCCURS.
6668                                      ;-
6669 034220  012767  004543  147542  4$:  MOV    #2403.,ERRNBR   ;SET ERROR NUMBER TO 2403.
6670 034226  012777  100012  146012       MOV    #100012,@TXCHA  ;WRITE THE DATA WORD TO THE DUT'S TXCHAR REG.
6671 034234  012701  170002              MOV    #170002,R1      ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
6672 034240  016702  146000              MOV    CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6673 034244  004767  167010              JSR    PC,WAIBIS       ;WAIT FOR TX_ACTION TO COME BACK.
6674 034250  103030                        BCC    6$              ;GO REPORT ERROR IF NO TX-ACTION FOUND.
6675                                      ;+
6676                                      ; WAIT FOR THE DATA TO APPEAR IN THE FIFO, REPORT ERROR IF TIME-OUT.
6677                                      ;-
6678 034252  005267  147512                INC    ERRNBR          ;SET ERROR NUMBER TO 2404.
6679 034256  012701  070012                MOV    #70012,R1       ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
6680 034262  016702  145756                MOV    CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6681 034266  004767  166766                JSR    PC,WAIBIS       ;WAIT FOR RX_DATA_AVAILABLE TO SET.
6682 034272  103017                        BCC    6$              ;REPORT ERROR IF NO DATA RECEIVED IN THE FIFO.
6683 034274  005267  147470                INC    ERRNBR          ;SET ERROR NUMBER TO 2405.
6684 034300  017702  145742                MOV    @RBUFA,R2       ;READ THE DATA FROM THE FIFO.
6685 034304  100012                        BPL    6$              ;GO REPORT ERROR IF THER IS'NT ANY DATA THERE.
6686 034306  005267  147456                INC    ERRNBR          ;SET ERROR NUMBER TO 2406.
6687 034312  000302                        SWAB   R2              ;PUT THE LINE NUMBER IN THE LOW BYTE.
6688 034314  042702  177760                BIC    #177760,R2      ;CLEAR THE UNWANTED BITS.
6689 034320  020204                        CMP    R2,R4           ;DID THE DATA COME FROM THE CORRECT LINE?.
6690 034322  001003                        BNE    6$              ;NO; GO REPORT THE ERROR.
6691 034324  005305                        DEC    R5              ;DECREMENT THE TXCHAR/LOOP COUNTER.
6692 034326  001334                        BNE    4$              ;LOOP TO TX THE NEXT CHAR.
6693 034330  000404                        BR     8$              ;GO TEST THE NEXT LINE.
6694
6695 034332  010401              6$:       MOV    R4,R1           ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6696 034334  012702  013457                MOV    #EM2302,R2      ;PASS THE MESSAGE TO BE REPORTED.
6697                                                              ;"TX_ENABLE BIT BAD ON LINE: NN".
6698 034340                                ERROR                  ;                >>>>> ERROR <<<<<.
     034340  104460                                                                    TRAP    C$ERROR
6699                                      ;+
6700                                      ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6701                                      ;-
6702 034342  010305              8$:       MOV    R3,R5           ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6703 034344  004767  166224                JSR    PC,TXDSBL       ;CLEAR THE TX_ENABLE BIT ON THIS LINE.
6704 034350  000241                        CLC                    ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
6705 034352  006103                        ROL    R3              ;SHIFT THE BIT MAP FOR THE NEXT LINE.
6706 034354  005204                        INC    R4              ;INCREMENT THE LINE NUMBER COUNTER.
6707 034356  020427  000010                CMP    R4,#NUMLNS      ;HAVE ALL THE LINES BEEN TESTED?.
6708 034362  002676                        BLT    2$              ;NO; BRANCH TO TEST THE NEXT LINE.
6709
6710 034364  005067  145762          60$:      CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6711 034370                                ENDTST
     034370                                                                  L10054:
     034370  104401                                                                  TRAP    C$ETST
```

```
6713                                    .SBTTL  HARDWARE TEST           - INTA -
6714                                    ;** ************************************************************************
6715                                    ;*                      - INTERRUPT TEST -
6716                                    ;*      THIS TEST VERIFIES THAT THE DEVICE UNDER TEST (DUT) WILL GENERATE
6717                                    ;*      RECEPTION AND TRANSMISSION INTERRUPTS CORRECTLY.  THIS TEST DOES
6718                                    ;*      NOT DEPEND ON THE USE OF THE SERIAL LINE TRANSMISSION OR RECEPTION
6719                                    ;*      CAPABILITIES OF THE DUT.  THE LINES ARE PUT IN INTERNAL LOOPBACK
6720                                    ;*      TO MINIMIZE ANY EXTERNAL EFFECTS THAT COULD BE CAUSED ON DEVICES
6721                                    ;*      ATTACHED TO THE SERIAL LINES.
6722                                    ;*
6723                                    ;-- ************************************************************************
6724
6725 034372                                    BGNTST
        034372                                                                          T24::
6726 034372                                    SETPRI  #PRIO5          ;ALLOW THE LTC TO INTERRUPT.
        034372  012700  000240                                                          MOV     #PRIO5,R0
        034376  104441                                                                  TRAP    C$SPRI
6727            000030                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6728 034400    012767  000030  145720           MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (26)
6729 034406    012767  177777  145736           MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
6730 034414    012767  000001  147344           MOV     #1,ERRTYP       ;SET ERROR FATAL ERROR TYPE IN ERROR TABLE.
6731 034422    012767  003101  147340           MOV     #1601.,ERRNBR   ;SET FIRST ERROR REPORT NUMBER IN ERROR TABLE.
6732 034430    012767  013551  147334           MOV     #EM2601.,ERRMSG ;SET TEST ERROR MESSAGE IN ERROR TABLE.
6733                                    ;*
6734                                    ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6735                                    ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6736                                    ; THIS SUBROUTINE REPORTS ERRORS FROM >>>>> 2601 THRU 2602 <<<<<.
6737                                    ;-
6738 034436    004767  165012                   JSR     PC,RESETT       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
6739 034442    103402                          BCS     2$              ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
6740 034444    000167  000740                  JMP     60$             ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
6741 034450    012767  005053  147312  2$:      MOV     #2603.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 2603.
6742                                    ;*
6743                                    ; ENABLE TRANSMITTERS ON ALL LINES.
6744                                    ;-
6745 034456    012705  000377          4$:      MOV     #MAPLNS,R5      ;PASS ACTIVE LINE BIT MAP.
6746 034462    004767  166202                   JSR     PC,TXENBL       ;ENABLE TRANSMISSION ON ALL LINES.
6747                                    ;*
6748                                    ; TEST RECEPTION INTERRUPTS.
6749                                    ;    SET UP FOR RX AND TX INTERRUPTS:
6750                                    ;    RX INTERRUPT SERVICE ROUTINE INPUTS A CHAR AND COUNTS THE INTERRUPT.
6751                                    ;    TX INTERRUPT SERVICE ROUTINE COUNTS TX INTERRUPTS.
6752                                    ;-
6753 034466    005067  145642                   CLR     RXINTC          ;CLEAR THE RX INTERRUPT COUNTER.
6754 034472    005067  145640                   CLR     RXINTF          ;CLEAR THE RX INTERRUPT FLAGS.
6755 034476    005067  145636                   CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNTER.
6756 034502    012767  002726  145614           MOV     #BUFBAS,BUFPTR  ;LOAD THE BUFFER PTR WITH THE BUFFER BASE ADR.
6757 034510                                     SETVEC  RXVECA,#RXINPT,#PRIO5 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
        034510  012746  000240                                                          MOV     #PRIO5,-(SP)
        034514  012746  024050                                                          MOV     #RXINPT,-(SP)
        034520  016746  145506                                                          MOV     RXVECA,-(SP)
        034524  012746  000003                                                          MOV     #3,-(SP)
        034530  104437                                                                  TRAP    C$SVEC
        034532  062706  000010                                                          ADD     #10,SP
6758 034536                                     SETVEC  TXVECA,#CACHTX,#PRIO5 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
        034536  012746  000240                                                          MOV     #PRIO5,-(SP)
        034542  012746  023662                                                          MOV     #CACHTX,-(SP)
```

HARDWARE TEST - INTA -

```
                034546  016746  145462                                                          MOV     TXVECA,-(SP)
                034552  012746  000003                                                          MOV     #3,-(SP)
                034556  104437                                                                  TRAP    C$SVEC
                034560  062706  000010                                                          ADD     #10,SP
        6759    034564
                034564  012700  000140                                                          MOV     #PRI03,R0
                034570  104441                                                                  TRAP    C$SPRI
        6760                                    ;+
        6761                                    ;ENABLE RECEPTION INTERRUPTS.
        6762                                    ;DELAY 4 MS TO ALLOW TIME FOR THE INTERRUPTS TO TAKE PLACE.
        6763                                    ;DISABLE RECEPTION INTERRUPTS.
        6764                                    ;-
        6765    034572  004767  165444                JSR     PC,RXIE1        ;ENABLE THE RECEPTION INTERRUPTS.
        6766    034576  012704  000004                 MOV    #4,R4           ;PASS 4 MS COUNT TO THE DELAY ROUTINE.
        6767    034602  004767  163470                JSR     PC,DELAY        ;DELAY 4 MILLI-SECONDS.
        6768    034606  004767  165370                JSR     PC,RXIE0        ;DISABLE RECEPTION INTERRUPTS.
        6769                                    ;+
        6770                                    ; VERIFY THAT THE CORRECT INTERRUPTS TOOK PLACE.
        6771                                    ;    TEST THE INT COUNTER TO VERIFY THAT INTERRUPTS TOOK PLACE.
        6772                                    ;-
        6773    034612  005767  145516                 TST    RXINTC          ;CHECK THE RX INTERRUPT COUNT.
        6774    034616  001017                BNE     6$              ;SKIP THE FOLLOWING ERRORS IF COUNT <> 0.
        6775                                    ;+
        6776                                    ; DETERMINE REASON FOR NO RX INTERRUPTS AND PRINT PROPER ERROR MESSAGE.
        6777                                    ;-
        6778    034620  012701  013760                 MOV    #EM2604,R1      ;SET UP MSG IN CASE "RX.DATA.AVAIL IS CLR".
        6779    034624  032777  000200  145412         BIT    #BIT7,@CSRA     ;TEST THE RX.DATA.AVAIL BIT OF THE CSR.
        6780    034632  001416                BEQ     8$              ;GO REPORT ERROR IF RX.DATA.AVAIL IS CLR.
        6781    034634  012701  013672                 MOV    #EM2603,R1      ;SET UP MSG IN CASE "DATA.VALID IS CLEAR".
        6782    034640  032777  100000  145400         BIT    #BIT15,@RBUFA   ;TEST THE DATA.VALID BIT OF THE FIFO.
        6783    034646  001410                BEQ     8$              ;GO REPORT ERROR IF DATA.VALID IS CLEAR.
        6784    034650  012701  013601                 MOV    #EM2602,R1      ;SET UP MSG,"DATA.VALID IS SET".
        6785    034654  000405                BR      8$              ;GO REPORT THE ERROR.
        6786                                    ;+
        6787                                    ; IF RX INTS OCCURRED WITH RX.DATA.AVAIL CLEAR, REPORT THE ERROR.
        6788                                    ;-
        6789    034656  005767  145454        6$:      TST    RXINTF          ;CHECK THE RX INTERRUPT FLAGS.
        6790    034662  100006                BPL     10$             ;SKIP THE ERROR IF FLAG IS CLEAR.
        6791    034664  012701  014054                 MOV    #EM2605,R1      ;SET UP THE PROPER MESSAGE.
        6792                                    ;+
        6793                                    ; REPORT THE ERROR WHICH HAS BEEN FOUND.
        6794                                    ;-
        6795    034670                          8$:      ERRDF  2603,EM2601,ER0503;             >>>>> ERROR #2603 <<<<<.
                034670  104455                                                                  TRAP    C$ERDF
                034672  005053                                                                  .WORD   2603
                034674  013551                                                                  .WORD   EM2601
                034676  016466                                                                  .WORD   ER0503
        6796                                    ;+
        6797                                    ; VERIFY THAT NO TX INTERRUPTS HAVE BEEN GENERATED SO FAR IN THIS TEST.
        6798                                    ;-
        6799    034700  016702  145434        10$:     MOV    TXINTC,R2       ;LOAD # OF TX INTERRUPTS FOR ER0504 RTN.
        6800    034704  001406                BEQ     12$             ;SKIP ERROR IF NO TX INTERRUPTS.
        6801                                ;REPORT "TX INTERRUPTS(S) RECEIVED WITH TX INTERRUPTS DISABLED."
        6802    034706  012701  010001                 MOV    #EM0526,R1      ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
        6803    034712                              ERRDF  2604,EM2601,ER0504;             >>>>> ERROR #2604 <<<<<.
                034712  104455                                                                  TRAP    C$ERDF
                034714  005054                                                                  .WORD   2604
```

```
      034716  013551                                                        .WORD   EM2601
      034720  016512                                                        .WORD   ER0504
 6804                                 ;+
 6805                                 ; CLEAN OUT THE INTERRUPT VECTORS USED IN THIS TEST.
 6806                                 ;-
 6807 034722                          12$:    SETPRI  #PRIO5          ;DISABLE DEVICE INTERRUPTS.
      034722  012700  000240                                                MOV     #PRIO5,R0
      034726  104441                                                      TRAP    C$SPRI
 6808 034730                                  CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
      034730  016700  145276                                              MOV     RXVECA,R0
      034734  104436                                                      TRAP    C$CVEC
 6809 034736                                  CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
      034736  016700  145272                                              MOV     TXVECA,R0
      034742  104436                                                      TRAP    C$CVEC
 6810                                 ;+
 6811                                 ; TEST TRANSMISSION INTERRUPTS.
 6812                                 ;    SET UP FOR RX AND TX INTERRUPTS:
 6813                                 ;    RX INTERRUPT SERVICE ROUTINE COUNTS RX INTERRUPTS.
 6814                                 ;    TX INTERRUPT SERVICE ROUTINE COUNTS THE INTERRUPT AND SETS FLAGS.
 6815                                 ;-
 6816 034744  005067  145364                   CLR     RXINTC          ;CLEAR THE RX INTERRUPT COUNTER.
 6817 034750  005067  145364                   CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNTER.
 6818 034754  005067  145362                   CLR     TXINTF          ;CLEAR THE RX INTERRUPT FLAGS.
 6819 034760                                   SETVEC  RXVECA,#CACHRX,#PRIO5 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
      034760  012746  000240                                                MOV     #PRIO5,-(SP)
      034764  012746  023634                                                MOV     #CACHRX,-(SP)
      034770  016746  145236                                                MOV     RXVECA,-(SP)
      034774  012746  000003                                                MOV     #3,-(SP)
      035000  104437                                                       TRAP    C$SVEC
      035002  062706  000010                                                ADD     #10,SP
 6820 035006                                   SETVEC  TXVECA,#TXINTR,#PRIO5 ;SET UP INT VECTOR TO TX INT ROUTINE.
      035006  012746  000240                                                MOV     #PRIO5,-(SP)
      035012  012746  024156                                                MOV     #TXINTR,-(SP)
      035016  016746  145212                                                MOV     TXVECA,-(SP)
      035022  012746  000003                                                MOV     #3,-(SP)
      035026  104437                                                       TRAP    C$SVEC
      035030  062706  000010                                                ADD     #10,SP
 6821 035034                                   SETPRI  #PRIO3          ;ALLOW DEVICE INTERRUPTS.
      035034  012700  000140                                                MOV     #PRIO3,R0
      035040  104441                                                       TRAP    C$SPRI
 6822                                 ;+
 6823                                 ; VERIFY THAT THE TX_ACTION BIT IS CLEAR.
 6824                                 ;-
 6825 035042  012705  000022                    MOV     #18.,R5         ;INITIALIZE THE LOOP COUNTER.
 6826 035046  012701  000144                    MOV     #100.,R1        ;SET 100 MS TIME-OUT.
 6827 035052  012702  100000                    MOV     #BIT15,R2       ;SELECT TX_ACTION BIT TO TEST.
 6828 035056  016704  145162                    MOV     CSRA,R4         ;PASS OUT CSR AS THE WORD TO TEST.
 6829 035062  012703  100000          14$:      MOV     #BIT15,R3       ;WAIT FOR TX_ACTION TO BE SET.
 6830 035066  004767  163360                    JSR     PC,MSLOOP       ;WAIT UP TO 100 MS FOR TX_ACTION SET.
 6831 035072  103020                            BCC     20$             ;IF TIME-OUT, CONSIDER TX_ACTION CLEAR.
 6832 035074  005003                             CLR     R3              ;NOW, WAIT FOR TX_ACTION CLEAR.
 6833 035076  004767  163350                    JSR     PC,MSLOOP       ;WAIT UP TO 100 MS FOR TX_ACTION CLEAR.
 6834 035102  103005                            BCC     16$             ;IF TIME-OUT, REPORT TX_ACTION WON'T CLEAR.
 6835 035104  005305                             DEC     R5              ;DECREMENT THE TX_ACTION SET COUNTER.
 6836 035106  001365                            BNE     14$             ;LOOP IF NOT TOO MANY TX_ACTIONS FOUND.
 6837                                 ;REPORT "TX_ACTION SET REPEATEDLY AFTER RESET, NO DATA SENT."
 6838 035110  012701  014176                    MOV     #EM2607,R1      ;SELECT ERROR MESSAGE.
```

```
HARDWARE TEST                - INTA -

 6839 035114  000402                          BR      18$             ;GO TO REPORT THE ERROR.
 6840 035116  012701  014272          16$:     MOV     #EM2608,R1      ;SELECT TX_ACTION STUCK SET MSG.
 6841 035122                          18$:     ERRDF   2605,EM2606,ER0503;             >>>>> ERROR #2605 <<<<<.
      035122  104455                                                                      TRAP    C$ERDF
      035124  005055                                                                      .WORD   2605
      035126  014137                                                                      .WORD   EM2606
      035130  016466                                                                      .WORD   ER0503
 6842 035132  000424                          BR      24$             ;GO TO TEST WITH TX_ACTION SET.
 6843                                 ;+
 6844                                 ; VERIFY THAT NO INTERRUPTS OCCUR WITH TX_ACTION CLEAR.
 6845                                 ;-
 6846 035134  004767  165664          20$:     JSR     PC,TXIE1        ;ENABLE TX_INTERRUPTS.
 6847 035140  012704  000062                   MOV     #50.,R4         ;PASS 50 MS TIME TO THE DELAY ROUTINE.
 6848 035144  004767  163126                   JSR     PC,DELAY        ;DELAY 50 MILLI-SECONDS TO ALLOW INTS TO OCCUR.
 6849 035150  005767  145164                   TST     TXINTC          ;TEST THE TX INTERRUPT COUNT.
 6850 035154  001413                          BEQ     24$             ;SKIP THE ERROR IF NO TX INTERRUPTS.
 6851 035156  012701  014176                   MOV     #EM2607,R1      ;SELECT MESSAGE IN CASE TX INT FLAG CLEAR.
 6852 035162  005767  145154                   TST     TXINTF          ;TEST THE TX INTERRUPT FLAGS.
 6853 035166  100002                          BPL     22$             ;GO REPORT ERROR IF TX FLAG IS CLEAR.
 6854 035170  012701  014343                   MOV     #EM2609,R1      ;TX FLAG IS SET, SELECT PROPER ERROR MESSAGE.
 6855                                 ;REPORT "TRANSMIT INTERRUPT TEST ERROR:..."
 6856 035174                          22$:     ERRDF   2606,EM2606,ER0503;             >>>>> ERROR #2606 <<<<<.
      035174  104455                                                                      TRAP    C$ERDF
      035176  005056                                                                      .WORD   2606
      035200  014137                                                                      .WORD   EM2606
      035202  016466                                                                      .WORD   ER0503
 6857                                 ;+
 6858                                 ; PREPARE TX INTERRUPT COUNTER AND FLAGS.
 6859                                 ;-
 6860 035204  005067  145130          24$:     CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNT.
 6861 035210  005067  145126                   CLR     TXINTF          ;CLEAR THE TX INTERRUPT FLAGS.
 6862                                 ;+
 6863                                 ; SET UP LINE PARAMETERS FOR TRANSMISSION.
 6864                                 ;-
 6865 035214  012705  000377                   MOV     #MAPLNS,R5      ;PASS ACTIVE LINES BIT MAP.
 6866 035220  012700  000200                   MOV     #200,R0         ;PASS INERT STATE, INTERNAL LOOPBACK.
 6867 035224  004767  166300                   JSR     PC,WTWLNC       ;DISABLE RECPTION AND DMA, ETC. ON DUT.
 6868 035230  012700  156430                   MOV     #156430,R0      ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
 6869 035234  004767  166344                   JSR     PC,WTWLPR       ;WRITE TO ALL LPR REGISTERS.
 6870                                 ;+
 6871                                 ; SEND A NULL CHAR TO EACH LINE.
 6872                                 ;-
 6873 035240  016701  145002                   MOV     TXCHA,R1        ;SET UP TXCHAR REGISTER ADDRESS.
 6874 035244  012702  100000                   MOV     #100000,R2      ;SET CHARACTER TO BE TRANSMITTED = NULL.
 6875 035250  004767  166304                   JSR     PC,WTWLNS       ;SEND NULL CHAR TO EACH LINE.
 6876                                 ;+
 6877                                 ; DELAY 250 MILLI-SECONDS TO ALLOW INTERRUPTS TO OCCUR.
 6878                                 ;-
 6879 035254  012704  000372                   MOV     #250.,R4        ;SET UP FOR 250 MS DELAY.
 6880 035260  004767  163012                   JSR     PC,DELAY        ;WAIT 250 MS.
 6881                                 ;+
 6882                                 ; VERIFY THAT TX INTERRUPTS OCCURRED.
 6883                                 ;-
 6884 035264  005767  145050                   TST     TXINTC          ;CHECK THE TX INTERRUPT COUNTER.
 6885 035270  001007                          BNE     26$             ;SKIP THE FOLLOWING ERROR IF WE GOT TX INTS.
 6886                                 ;+
 6887                                 ; DETERMINE THE REASON THAT WE RECEIVED NO INTERRUPTS.
```

```
 6888
 6889 035272  012701  014422                ;-
                                                MOV     #EM2610,R1      ;SET UP MSG IN CASE "TX_ACTION IS SET".
 6890 035276  005777  144742                    TST     @CSRA           ;CHECK THE DUT CSR.
 6891 035302  100407                          BMI     28$             ;GO TO REPORT ERROR IF TX_ACTION IS SET.
 6892 035304  012701  014514                    MOV     #EM2611,R1      ;SET UP "TX_ACTION NOT SET" MESSAGE.
 6893                                        ;*
 6894                                        ; CHECK TO VERIFY THAT TX_ACTION WAS SET FOR EACH INTERRUPT.
 6895                                        ;-
 6896 035310  005767  145026          26$:      TST     TXINTF          ;CHECK THE TX INTERRUPT FLAGS.
 6897 035314  100006                          BPL     30$             ;SKIP ERROR IF TX_ACTION CLR FLAG IS CLEAR.
 6898 035316  012701  014343                    MOV     #EM2609,R1      ;SET UP TX INT WITH "TX_ACTION CLR" MSG.
 6899                                        ;*
 6900                                        ; REPORT "TRANSMIT INTERRUPT TEST ERROR:...."
 6901                                        ;-
 6902 035322                          28$:      ERRDF   2607,EM2606,ER0503;             >>>>> ERROR #2607 <<<<<.
      035322  104455                                                                       TRAP    C$ERDF
      035324  005057                                                                       .WORD   2607
      035326  014137                                                                       .WORD   EM2606
      035330  016466                                                                       .WORD   ER0503
 6903                                        ;*
 6904                                        ; VERIFY THAT NO TX INTERRUPTS HAVE BEEN GENERATED SO FAR IN THIS TEST.
 6905                                        ;-
 6906 035332  016702  144776          30$:      MOV     RXINTC,R2       ;LOAD # OF RX INTERRUPTS FOR ER0504 RTN.
 6907 035336  001406                          BEQ     32$             ;SKIP ERROR IF NO RX INTERRUPTS.
 6908 035340  012701  007711                    MOV     #EM0525,R1      ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
 6909                                       ;REPORT "RX INTERRUPTS(S) RECEIVED WITH RX INTERRUPTS DISABLED."
 6910 035344                                    ERRDF   2608,EM2606,ER0504;             >>>>> ERROR #2608 <<<<<.
      035344  104455                                                                       TRAP    C$ERDF
      035346  005060                                                                       .WORD   2608
      035350  014137                                                                       .WORD   EM2606
      035352  016512                                                                       .WORD   ER0504
 6911                                        ;*
 6912                                        ; DISABLE INTERRUPTS AND CLEAN OUT THE INTERRUPT VECTORS USED IN THIS TEST.
 6913                                        ;-
 6914 035354  005001                  32$:      CLR     R1              ;CLEAR BOTH TRANSMITTER
 6915 035356  004767  165402                    JSR     PC,TXIE0        ; INTERRUPT ENABLE AND RECEIVER
 6916 035362  004767  164614                    JSR     PC,RXIE0        ; INTERRUPT ENABLE BITS IN THE DUT CSR.
 6917 035366                                    SETPRI  #PRIO5          ;DISABLE DEVICE INTERRUPTS.
      035366  012700  000240                                                                MOV     #PRIO5,R0
      035372  104441                                                                       TRAP    C$SPRI
 6918 035374                                    CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
      035374  016700  144632                                                                MOV     RXVECA,R0
      035400  104436                                                                       TRAP    C$CVEC
 6919 035402                                    CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
      035402  016700  144626                                                                MOV     TXVECA,R0
      035406  104436                                                                       TRAP    C$CVEC
 6920
 6921 035410  005067  144736          60$:      CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 6922 035414                                    SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
      035414  012700  000340                                                                MOV     #PRIO7,R0
      035420  104441                                                                       TRAP    C$SPRI
 6923 035422                                    ENDTST
      035422                                                                          L10055:
      035422  104401                                                                       TRAP    C$ETST
 6924
```

```
                                .SBTTL  HARDWARE TEST           - BRLEVA -
                                ;** ****************************************************************************
                                ;*                          - BR LEVEL TEST B -
                                ;*      THIS TEST VERIFIES THAT THE DEVICE UNDER TEST (DUT) WILL GENERATE
                                ;*      RECEPTION AND TRANSMISSION INTERRUPTS AT THE CORRECT BR LEVEL.
                                ;*      THIS TEST DOES NOT DEPEND ON THE USE OF THE SERIAL LINE TRANSMISSION
                                ;*      OR RECEPTION CAPABILITIES OF THE DUT.  THE LINES ARE PUT IN INTERNAL
                                ;*      LOOPBACK TO MINIMIZE ANY EXTERNAL EFFECTS THAT COULD BE CAUSED ON
                                ;*      DEVICES ATTACHED TO THE SERIAL LINES.
                                ;*
                                ;-- ****************************************************************************

035424                                  BGNTST
                                                                                T25::
035424                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
035424  012700  000240                                                          MOV     #PRIO5,R0
035430  104441                                                                  TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
035432  012767  000031  144666          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (30)
035440  012767  177777  144704          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
035446  012767  000001  146312          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
035454  012767  005671  146306          MOV     #3001.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
035462  012767  014577  146302          MOV     #EM3001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
035470  005067  144766                  CLR     ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS FROM >>>>> 3001 THRU 3002 <<<<<.
                                ;-
035474  004767  163754                  JSR     PC,RESETT       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
035500  103402                          BCS     2$              ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
035502  000167  000572                  JMP     60$             ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
035506  012767  005673  146254  2$:      MOV     #3003.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 3003.
                                ;*
                                ; ENABLE TRANSMITTERS ON ALL LINES.
                                ;-
035514  012705  000377          4$:      MOV     #MAPLNS,R5      ;PASS ACTIVE LINE BIT MAP.
035520  004767  165144                  JSR     PC,TXENBL       ;ENABLE TRANSMISSION ON ALL LINES.
                                ;*
                                ; GENERATE A TRANSMISSION INTERRUPT REQUEST.
                                ; PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING INTS.
                                ;-
035524                                  SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
035524  012700  000340                                                          MOV     #PRIO7,R0
035530  104441                                                                  TRAP    C$SPRI
035532                                  SETVEC  TXVECA,#TXINTR,#PRIO7 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
035532  012746  000340                                                          MOV     #PRIO7,-(SP)
035536  012746  024156                                                          MOV     #TXINTR,-(SP)
035542  016746  144466                                                          MOV     TXVECA,-(SP)
035546  012746  000003                                                          MOV     #3,-(SP)
035552  104437                                                                  TRAP    C$SVEC
035554  062706  000010                                                          ADD     #10,SP
                                ;*
                                ; SET UP DUT FOR TRANSMISSION INTERRUPTS:
                                ;  SET UP INTERNAL LOOPBACK.
                                ;  SET UP LINE PARAMETERS FOR TRANSMISSION.
                                ;-
```

HARDWARE TEST - BRLEVA -

```
 6972 035560  012705  000377          MOV     #MAPLNS,R5      ;PASS ACTIVE LINES BIT MASK.
 6973 035564  012700  000200          MOV     #200,R0         ;PASS INERT STATE, INTERNAL LOOPBACK.
 6974 035570  004767  165734          JSR     PC,WTWLNC       ;DISABLE RECPTION AND DMA, ETC. ON DUT.
 6975 035574  012700  156430          MOV     #156430,R0      ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
 6976 035600  004767  166000          JSR     PC,WTWLPR       ;WRITE INTO ALL LPR REGISTERS.
 6977                                ;+
 6978                                ; SEND A NULL CHAR TO EACH LINE.
 6979                                ;-
 6980 035604  016701  144436          MOV     TXCHA,R1        ;SET UP TXCHAR REGISTER ADDRESS.
 6981 035610  012702  100000          MOV     #100000,R2      ;SET CHARACTER TO BE TRANSMITTED = NULL.
 6982 035614  004767  165740          JSR     PC,WTWLNS       ;SEND NULL CHAR TO EACH LINE.
 6983                                ;+
 6984                                ; DELAY 50 MS TO ALLOW TIME FOR THE INTERRUPT TO BE GENERATED.
 6985                                ;-
 6986 035620  012704  000062          MOV     #50.,R4         ;PASS 50 MS TIME TO THE DELAY ROUTINE.
 6987 035624  004767  162446          JSR     PC,DELAY        ;DELAY 50 MILLI-SECONDS.
 6988                              ;+
 6989                              ; GENERATE A RECEPTION INTERRUPT REQUEST.
 6990                              ;-
 6991 035630                            SETVEC  RXVECA,#RXBRRT,#PRIO7 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
      035630  012746  000340                                                 MOV     #PRIO7,-(SP)
      035634  012746  023760                                                 MOV     #RXBRRT,-(SP)
      035640  016746  144366                                                 MOV     RXVECA,-(SP)
      035644  012746  000003                                                 MOV     #3,-(SP)
      035650  104437                                                        TRAP    C$SVEC
      035652  062706  000010                                                 ADD     #10,SP
 6992                              ;+
 6993                              ; SET UP FOR THE LOOP WHICH TESTS THE INTERRUPT BR LEVELS.
 6994                              ;-
 6995 035656  012705  000340          MOV     #340,R5         ;SET UP THE PRIORITY LEVEL TO 7.
 6996 035662  005003                  CLR     R3              ;CLEAR THE RX PRIORITY STORE AND FLAGS.
 6997 035664  005002                  CLR     R2              ;CLEAR THE TX PRIORITY STORE AND FLAGS.
 6998                                ;+
 6999                                ; ENABLE TX AND RX INTERRUPTS.
 7000                                ; PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING THE INTERRUPTS.
 7001                                ;-
 7002 035666  004767  164350          JSR     PC,RXIE1        ;ENABLE RECEIVER INTERRUPTS.
 7003 035672  004767  165126          JSR     PC,TXIE1        ;ENABLE TRANSMITTER INTERRUPTS.
 7004                              ;+
 7005                              ; LOOP, LOWERING THE PROCESSOR PRIORITY UNTIL THE DUT INTERRUPTS ON RX AND TX.
 7006                              ;-
 7007 035676  005067  144436  6$:       CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNTER.
 7008 035702  005067  144434           CLR     TXINTF          ;CLEAR THE TX INTERRUPT FLAGS.
 7009 035706  005067  144422           CLR     RXINTC          ;CLEAR THE RX INTERRUPT COUNTER.
 7010 035712  005067  144420           CLR     RXINTF          ;CLEAR THE RX INTERRUPT FLAGS.
 7011 035716                          SETPRI  R5              ;SET PROCESSOR PRIORITY TO THE SELECTED VALUE.
      035716  010500                                                 MOV     R5,R0
      035720  104441                                                 TRAP    C$SPRI
 7012 035722  012704  000001          MOV     #1,R4           ;PASS 1 MS COUNT TO THE DELAY ROUTINE.
 7013 035726  004767  162344          JSR     PC,DELAY        ;DELAY 1 MS TO ALLOW INTERRUPTS TO OCCUR.
 7014                                ;+
 7015                                ; DETERMINE IF ANY RX DUT INTERRUPTS OCCURRED.
 7016                                ; LOG THE PROCESSOR PRIORITY FOR THE RX INTERRUPT IF FIRST RX INT.
 7017                                ;-
 7018 035732  005767  144376          TST     RXINTC          ;CHECK THE RECEIVE INTERRUPT COUNTER.
 7019 035736  001412                  BEQ     8$              ;SKIP THE PRIORITY LOG IF NO RX INT OCCURRED.
 7020                                 ;+
```

```
7021                                    ; IF THIS IS THE FIRST RX INTERRUPT, LOG THE PRIORITY.
7022                                    ;-
7023 035740  005703                      TST    R3              ;CHECK THE RX PRIORITY STORE AND FLAGS.
7024 035742  001010                     BNE    8$              ;GOTO TEST FOR TX INTS IF NOT THE FIRST RX INT.
7025 035744  010503                      MOV    R5,R3           ;LOG THE PRESENT PRIORITY IN THE RX PRIO STORE.
7026 035746  052703  100000              BIS    #BIT15,R3       ;SET THE RX INT HAS OCCURRED FLAG.
7027 035752  016700  144360              MOV    RXINTF,R0       ;GET THE RX INTERRUPT ROUTINE FLAGS.
7028 035756  042700  137777              BIC    #137777,R0      ;CLEAR ALL BUT THE TX INT ERROR FLAG.
7029 035762  050003                      BIS    R0,R3           ;IF TX INT ERROR, SET BIT 14 OF THE PRIO FLAGS.
7030                                    ;+
7031                                    ; DETERMINE IF ANY TX DUT INTERRUPTS HAVE OCCURRED.
7032                                    ; LOG THE PRESENT PROCESSOR PRIORITY IF THIS IS THE FIRST TX INTERRUPT.
7033                                    ;-
7034 035764  005767  144350        8$:     TST    TXINTC          ;CHECK THE TRANSMIT INTERRUPT COUNTER.
7035 035770  001405                     BEQ    10$             ;SKIP THE PRIORITY LOG IF NO TX INT OCCURRED.
7036                                    ;+
7037                                    ; IF THIS IS THE FIRST TX INTERRUPT, LOG THE PRIORITY.
7038                                    ;-
7039 035772  005702                      TST    R2              ;CHECK THE TX PRIORITY STORE AND FLAGS.
7040 035774  100403                     BMI    10$             ;SKIP THE LOGGING IF NOT FIRST TX INTERRUPT.
7041 035776  010502                      MOV    R5,R2           ;LOG THE PRESENT PRIORITY IN THE TX PRIO STORE.
7042 036000  052702  100000              BIS    #BIT15,R2       ;SET THE TX INT HAS OCCURRED FLAG.
7043                                    ;+
7044                                    ; SELECT NEXT PROCESSOR PRIORITY.
7045                                    ; TEST FOR BOTH RX AND TX INTERRUPTS HAVING OCCURRED, LOOP IF NOT.
7046                                    ;-
7047 036004  162705  000040        10$:    SUB    #40,R5          ;DECREMENT PRIORITY LEVEL BY ONE.
7048 036010  002402                     BLT    12$             ;GOTO CHECK FOR ERRORS IF BELOW PRIORITY ZERO.
7049 036012  030203                      BIT    R2,R3           ;AND PRIO FLAGS TOGETHER, ALTER NONE OF THEM.
7050 036014  100330                     BPL    6$              ;LOOP IF RX AND TX INTS HAVEN'T BOTH OCCURRED.
7051                                    ;+
7052                                    ; DISABLE INTERRUPTS AND CLEAR INTERRUPT VECTORS.
7053                                    ;-
7054 036016                        12$:    SETPRI #PRI07          ;DISABLE ALL INTERRUPTS.
     036016  012700  000340                                                    MOV    #PRI07,R0
     036022  104441                                                           TRAP   C$SPRI
7055 036024                             CLRVEC RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
     036024  016700  144202                                                    MOV    RXVECA,R0
     036030  104436                                                           TRAP   C$CVEC
7056 036032                             CLRVEC TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
     036032  016700  144176                                                    MOV    TXVECA,R0
     036036  104436                                                           TRAP   C$CVEC
7057                                    ;+
7058                                    ; VERIFY THAT RX AND TX INTERRUPTS OCCURRED,
7059                                    ; AT THE PROPER BR LEVEL, AND
7060                                    ; IN THE PROPER ORDER.
7061                                    ;    DETERMINE IF TX INTERRUPT OCCURRED.
7062                                    ;-
7063 036040  005702                          TST    R2              ;DETERMINE WHETHER TX INT OCCURRED OR NOT.
7064 036042  100414                         BMI    16$             ;SKIP THESE ERRORS IF TX INT OCCURRED.
7065                                    ;+
7066                                    ; DETERMINE REASON THAT NO TX INT OCCURRED.
7067                                    ;-
7068 036044  012701  014422                  MOV    #EM2610,R1      ;SELECT "NO TX INT FROM TX.ACTION" MESSAGE.
7069 036050  005777  144170                  TST    @CSRA           ;CHECK THE TX.ACTION BIT OF THE DUT CSR.
7070 036054  100402                         BMI    14$             ;SKIP TX.ACTION CLR MSG SELECTION IF IT IS SET.
7071 036056  012701  014514                  MOV    #EM2611,R1      ;SELECT "TX.ACTION CLEAR AFTER CHARS SENT" MSG.
```

```
7072                                    ;REPORT "INTERRUPT BR LEVEL TEST ERROR:"
7073 036062                         14$:    ERRDF   3003,EM3001,ER0503;          >>>>> ERROR #3003 <<<<<.
     036062  104455                                                                       TRAP    C$ERDF
     036064  005673                                                                       .WORD   3003
     036066  014577                                                                       .WORD   EM3001
     036070  016466                                                                       .WORD   ER0503
7074 036072  000423                         BR      18$             ;SKIP THE BR LEVEL CHECK, NO TX INT OCCURRED.
7075                                    ;+
7076                                    ; VERIFY THAT THE TX INTERRUPT WAS AT THE PROPER BR LEVEL.
7077                                    ;-
7078 036074  010204                 16$:    MOV     R2,R4           ;CALCULATE THE BR LEVEL
7079 036076  042704  177400                BIC     #177400,R4      ; THAT THE TRANMSIT
7080 036102  006204                        ASR     R4              ; INTERRUPT WAS
7081 036104  006204                        ASR     R4              ; REQUESTED AT, WHICH
7082 036106  006204                        ASR     R4              ; IS ONE GREATER THAN
7083 036110  006204                        ASR     R4              ; THE PROCESSOR PRIORITY
7084 036112  006204                        ASR     R4              ; LEVEL AT WHICH THE
7085 036114  005204                        INC     R4              ; TRANSMIT INTERRUPT OCCURRED.
7086 036116  116705  144120              MOVB    BRLEVL,R5       ;GET THE EXPECTED INTERRUPT BR LEVEL.
7087 036122  120405                      CMPB    R4,R5           ;COMARE THE INTERRUPT BR LEVEL WITH EXPECTED.
7088 036124  001406                      BEQ     18$             ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
7089                                    ;REPORT "TX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
7090 036126  012701  014724              MOV     #EM3003,R1      ;SELECT THE ERROR MESSAGE FOR THE ERROR CALL.
7091 036132                             ERRDF   3004,EM3001,ER3001;          >>>>> ERROR #3004 <<<<<.
     036132  104455                                                                       TRAP    C$ERDF
     036134  005674                                                                       .WORD   3004
     036136  014577                                                                       .WORD   EM3001
     036140  017066                                                                       .WORD   ER3001
7092                                    ;+
7093                                    ; DETERMINE IF RX INTERRUPT OCCURRED.
7094                                    ;-
7095 036142  005703                 18$:    TST     R3              ;CHECK THE RX INT OCCURRED FLAG.
7096 036144  100415                        BMI     22$             ;SKIP THESE ERRORS IF RX INT OCCURRED.
7097                                    ;+
7098                                    ; DETERMINE REASON THAT NO RX INT OCCURRED.
7099                                    ;-
7100 036146  012701  013601                MOV     #EM2602,R1      ;SELECT "NO RX INT FROM TX.ACTION" MSG.
7101 036152  032777  000200  144064        BIT     #BIT7,@CSRA     ;CHECK THE RX.DATA.AVAIL BIT OF THE DUT CSR.
7102 036160  001002                      BNE     20$             ;SKIP RX.DATA.AVAIL CLR MSG IF BIT IS SET.
7103 036162  012701  014630              MOV     #EM3002,R1      ;SELECT "NO RX.DATA.AVAIL AFTER RESET" MSG.
7104                                    ;REPORT "INTERRUPT BR LEVEL TEST ERROR:"
7105 036166                         20$:    ERRDF   3005,EM3001,ER0503;          >>>>> ERROR #3005 <<<<<.
     036166  104455                                                                       TRAP    C$ERDF
     036170  005675                                                                       .WORD   3005
     036172  014577                                                                       .WORD   EM3001
     036174  016466                                                                       .WORD   ER0503
7106 036176  000423                         BR      24$             ;SKIP THE BR CHECK IF NO RX INT OCCURRED.
7107                                    ;+
7108                                    ; VERIFY THAT THE RX INTERRUPT WAS AT THE PROPER BR LEVEL.
7109                                    ;-
7110 036200  010304                 22$:    MOV     R3,R4           ;CALCULATE THE BR LEVEL
7111 036202  042704  177400                BIC     #177400,R4      ; THAT THE RECEIVE
7112 036206  006204                        ASR     R4              ; INTERRUPT WAS
7113 036210  006204                        ASR     R4              ; REQUESTED AT, WHICH
7114 036212  006204                        ASR     R4              ; IS ONE GREATER THAN
7115 036214  006204                        ASR     R4              ; THE PROCESSOR PRIORITY
7116 036216  006204                        ASR     R4              ; LEVEL AT WHICH THE
```

```
7117 036220  005204                    INC     R4                  ; RECEIVE INTERRUPT OCCURRED.
7118 036222  116705  144014            MOVB    BRLEVL,R5           ;GET THE EXPECTED INTERRUPT BR LEVEL.
7119 036226  120405                    CMPB    R4,R5               ;COMARE THE INTERRUPT BR LEVEL WITH EXPECTED.
7120 036230  001406                    BEQ     24$                 ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
7121                                 ;REPORT "RX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
7122 036232  012701  015000            MOV     #EM3004,R1          ;SELECT ERROR MESSAGE FOR THE ERROR CALL.
7123 036236                            ERRDF   3006,EM3001,ER3001; >>>>> ERROR #3006 <<<<<.
     036236  104455                                                        TRAP    C$ERDF
     036240  005676                                                        .WORD   3006
     036242  014577                                                        .WORD   EM3001
     036244  017066                                                        .WORD   ER3001
7124
7125                                   ;*
7126                                   ; TEST FOR INTERRUPTS OCCURING IN THE PROPER ORDER.
7127                                   ;-
7128 036246  032703  040000        24$:    BIT     #BIT14,R3           ;CHECK THE IMPROPER INT ORDER ERROR FLAG.
7129 036252  001406                    BEQ     26$                 ;SKIP ERROR REPORT IF ERROR DID NOT OCCUR.
7130                                 ;REPORT "TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS RX INT."
7131 036254  012701  015054            MOV     #EM3005,R1          ;SELECT THE ERROR MESSAGE FOR INDIRECT PRINT.
7132 036260                            ERRDF   3007,EM3001,ER0503; >>>>> ERROR #3007 <<<<<.
     036260  104455                                                        TRAP    C$ERDF
     036262  005677                                                        .WORD   3007
     036264  014577                                                        .WORD   EM3001
     036266  016466                                                        .WORD   ER0503
7133                                   ;*
7134                                   ; CLEAN UP, EXIT THE TEST.
7135                                   ;-
7136 036270  004767  164470        26$:    JSR     PC,TXIE0            ;CLEAR TRANSMITTER INTERRUPTS.
7137 036274  004767  163702            JSR     PC,RXIE0            ;CLEAR RECEIVER INTERRUPTS.
7138 036300  005067  144046        60$:    CLR     CTRLCF              ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7139 036304                            SETPRI  #PRI07              ;DISABLE ALL INTERRUPTS.
     036304  012700  000340                                                MOV     #PRI07,R0
     036310  104441                                                        TRAP    C$SPRI
7140 036312                            ENDTST
     036312                                                     L10056:
     036312  104401                                                        TRAP    C$ETST
7141
```

```
                                .SBTTL  HARDWARE TEST           - DIABMP -
                                ;** ************************************************************************
                                ;*                        - DIAGNOSTIC FIELD (BMP) TEST -
                                ;*      THIS TEST VERIFIES THAT A REQUEST TO THE DUT TO REPORT BMP STATUS
                                ;*      CODES IS COMPLIED WITH, WITHIN THE SPECIFIED TIME.
                                ;*      ALL ACTIVE LINES ARE TESTED.
                                ;-- ************************************************************************

                                        BGNTST
                                                                                        T26::
                                        SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
036314  012700  000240                                                          MOV     #PRI05,R0
036320  104441                                                                 TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
036322  012767  000032  143776         MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (31)
036330  012767  177777  144014         MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
036336  012767  000001  145422         MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
036344  012767  006035  145416         MOV     #3101.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
036352  012767  015146  145412         MOV     #EM3101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
036360  012767  017420  145406         MOV     #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 3101 <<<<<.
                                ;-
036366  004767  161562                 JSR     PC,CLNRST       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
036372  103077                         BCC     60$             ;RESET FAILURE?, ABORT THIS TEST.
                                ;+
                                ; TEST ALL ACTIVE LINES INDIVIDUALLY.
                                ; WRITE THE REQUEST CODE TO THE DIAGNOSTIC FIELD IN THE LPR REGISTER.
                                ; VERIFY THAT A BMP CODE IS RETURNED WITHIN THE CORRECT TIME.
                                ;-
036374  016705  143636                   MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
036400  005004                           CLR     R4              ;CLEAR THE LINE NUMBER COUNTER.
036402  016703  143636                   MOV     CSRA,R3         ;GET THE ADDRESS OF THE DUT'S CSR.
036406  000241                 2$:       CLC                     ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
036410  006005                           ROR     R5              ;SHIFT THE BIT MAP INTO THE CARRY BIT.
036412  103064                         BCC     8$              ;DO NOT TEST THE LINE IF IT IS INACTIVE.
                                    ;+
                                    ; SELECT THE LINE UNDER TEST.
                                    ;  WRITE THE BMP REQUEST CODE TO THE DIAG FIELD IN THE LPR REGISTER.
                                    ;-
036414  012767  006036  145346           MOV     #3102.,ERRNBR   ;SET THE ERROR NUMBER TO 3102.
036422  010413                           MOV     R4,(R3)         ;SELECT THE LINE CURRENTLY UNDER TEST.
036424  052777  000002  143616           BIS     #2,@LPRA        ;WRITE THE BMP REQUEST CODE TO THE LPR.
                                      ;+
                                      ; WAIT FOR BMP REQUEST CODE TO BE CLEARED, REPORT ERROR IF TIME-OUT
                                      ; OCCURS.
                                      ;-
036432  012701  010764                 MOV     #10764,R1       ;TEST BIT 1, TIMEOUT OF 500 MILLI SECS.
036436  016702  143606                 MOV     LPRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
036442  004767  164536                JSR     PC,WAIBIC       ;WAIT FOR REQUEST CODE TO CLEAR.
036446  103042                        BCC     6$              ;GO REPORT ERROR IF CODE DID NOT CLEAR IN TIME.
                                      ;+
                                      ; WAIT FOR BMP CODE TO APPEAR IN THE FIFO, REPORT ERROR IF TIME-OUT
                                      ; OCCURS.
                                      ;-
```

HARDWARE TEST - DIABMP -

```
7197 036450  005267  145314          INC     ERRNBR          ;SET ERROR NUMBER TO 3103.
7198 036454  012701  070012          MOV     #70012,R1       ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
7199 036460  016702  143560          MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7200 036464  004767  164570          JSR     PC,WAIBIS       ;WAIT FOR RX_DATA_AVAILABLE TO SET.
7201 036470  103031                  BCC     6$              ;GO REPORT ERROR IF CODE DID NOT CLEAR IN TIME.
7202                                 ;+
7203                                 ; READ THE BMP CODE (IF IT IS THERE) FROM THE RBUF REGISTER.
7204                                 ; DETERMINE IF IT IS A VALID BMP CODE.
7205                                 ; VERIFY THE BMP CODE WAS RECEIVED FROM THE CORRECT CHANNEL.
7206                                 ; IF THE BMP CODE DOES NOT INDICATE DUT RUNNING OK, THEN SAVE IT ON
7207                                 ; THE QUEUE TO BE REPORTED IN A LATER TEST.
7208                                 ;-
7209 036472  005267  145272          INC     ERRNBR          ;SET ERROR NUMBER TO 3104.
7210 036476  017702  143544          MOV     @RBUFA,R2       ;GET THE BMP CODE FROM THE FIFO.
7211 036502  100024                  BPL     6$              ;GO REPORT ERROR IF NO BMP CODE FOUND.
7212 036504  005267  145260          INC     ERRNBR          ;SET ERROR NUMBER TO 3105.
7213 036510  012700  170301          MOV     #170301,R0      ;SET-UP A BMP CODE MASK.
7214 036514  040200                  BIC     R2,R0           ;TRY TO CLEAR THE BMP MASK.
7215 036516  001016                  BNE     6$              ;GO REPORT ERROR IF IT IS NOT A VALID BMP CODE.
7216 036520  005267  145244          INC     ERRNBR          ;SET THE ERROR NUMBER TO 3106.
7217 036524  010200                  MOV     R2,R0           ;COPY THE BMP CODE.
7218 036526  000300                  SWAB    R0              ;PUT THE LINE NUMBER IN THE LOW BYTE.
7219 036530  042700  177760          BIC     #177760,R0      ;CLEAR THE UNWANTED BITS.
7220 036534  120400                  CMPB    R4,R0           ;DID THE BMP CODE COME FROM THE CORRECT LINE?.
7221 036536  001006                  BNE     6$              ;NO; GO REPORT ERROR.
7222 036540  120227  000305          CMPB    R2,#305         ;IS THE BMP CODE A "GOOD ONE"?.
7223 036544  001407                  BEQ     8$              ;YES; SKIP SAVING THE BMP CODE ON THE QUEUE.
7224 036546  004767  163514          JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
7225 036552  000404                  BR      8$              ;GO SEE IF THERE ARE ANY MORE LINE TO TEST.
7226
7227 036554  010401          6$:     MOV     R4,R1           ;PASS THE LINE NUMBER TO BE REPORTED.
7228 036556  012702  015202          MOV     #EM3102,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
7229                                                         ;"BMP REQUEST BIT BAD ON LINE:"
7230 036562                          ERROR                   ;               >>>>> ERROR <<<<<.
     036562  104460                                                              TRAP    C$ERROR
7231                                 ;+
7232                                 ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
7233                                 ;-
7234 036564  005204          8$:     INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
7235 036566  005705                  TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
7236 036570  001306                  BNE     2$              ;YES; BRANCH TO TEST THE NEXT LINE.
7237 036572  005067  143554  60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7238 036576                          ENDTST
     036576                                                                  L10057:
     036576  104401                                                              TRAP    C$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.
                                ;*
                                ;-- ************************************************************************
036600                                  BGNTST
036600                                                                          T27::
                 000033                  TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
036600  012767  000033  143520          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (93)
036606  012767  177777  143536          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
036614  016702  143704                  MOV     BMPCQP,R2       ;GET THE CONTENTS OF THE POINTER.
036620  012703  002526                  MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
036624  020203                          CMP     R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
036626  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"

036630  012701  016011                  MOV     #EM9304,R1      ;PASS THE FIRST MESSAGE TO BE REORTED.
036634                                  ERRDF   9301,EM9301,ER9301 ;            >>>>> ERROR #9301 <<<<<.
036634  104455                                                                  TRAP    C$ERDF
036636  022125                                                                  .WORD   9301
036640  015674                                                                  .WORD   EM9301
036642  017446                                                                  .WORD   ER9301

036644  012767  002526  143652          MOV     #BMPCQB,BMPCQP  ;SET POINTER BACK TO THE BEGINING OF THE QUE.

036652  005067  143474          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
036656                                  ENDTST
036656                                                                  L10060:
036656  104401                                                                  TRAP    C$ETST
```

HARDWARE TEST - REPEMP -

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

036660                          BGNHRD
036660  000022                                                    .WORD L10061-L$HARD/2
036662                                                          L$HARD::

                        ;DEVICE CSR ADDRESS QUESTION:
036662                          GPRMA   HWPTQ1,0,0,160000,177776,YES
036662  000031                                                  .WORD   T$CODE
036664  036726                                                  .WORD   HWPTQ1
036666  160000                                                  .WORD   T$LOLIM
036670  177776                                                  .WORD   T$HILIM
                        ;DEVICE INTERRUPT VECTOR QUESTION:
036672                          GPRMA   HWPTQ2,2,0,40,776,YES
036672  001031                                                  .WORD   T$CODE
036674  036744                                                  .WORD   HWPTQ2
036676  000040                                                  .WORD   T$LOLIM
036700  000776                                                  .WORD   T$HILIM
                        ;ACTIVE LINES BIT MAP QUESTION:
036702                          GPRMD   HWPTQ3,4,0,MAPLNS,0,177777,YES
036702  002032                                                  .WORD   T$CODE
036704  036777                                                  .WORD   HWPTQ3
036706  000377                                                  .WORD   MAPLNS
036710  000000                                                  .WORD   T$LOLIM
036712  177777                                                  .WORD   T$HILIM
                        ;INTERRUPT BR LEVEL QUESTION:
036714                          GPRMD   HWPTQ5,6,0,377,0,6,YES
036714  003032                                                  .WORD   T$CODE
036716  037025                                                  .WORD   HWPTQ5
036720  000377                                                  .WORD   377
036722  000000                                                  .WORD   T$LOLIM
036724  000006                                                  .WORD   T$HILIM


036726                          ENDHRD
                                                                .EVEN
036726                                                      L10061:
```

HARDWARE PARAMETER CODING SECTION

```
 7324 036726  103  123  122        HWPTQ1: .ASCIZ  /CSR ADDRESS: /
      036731  040  101  104
      036734  104  122  105
      036737  123  123  072
      036742  040  000
 7325 036744  111  116  124        HWPTQ2: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
      036747  105  122  122
      036752  125  120  124
      036755  040  126  105
      036760  103  124  117
      036763  122  040  101
      036766  104  104  122
      036771  105  123  123
      036774  072  040  000
 7326 036777  101  103  124        HWPTQ3: .ASCIZ  /ACTIVE LINE BIT MAP: /
      037002  111  126  105
      037005  040  114  111
      037010  116  105  040
      037013  102  111  124
      037016  040  115  101
      037021  120  072  040
      037024  000
 7327 037025  111  116  124        HWPTQ5: .ASCIZ  /INTERRUPT BR LEVEL: /
      037030  105  122  122
      037033  125  120  124
      037036  040  102  122
      037041  040  114  105
      037044  126  105  114
      037047  072  040  000
 7328
 7329                             .EVEN
```

HARDWARE PARAMETER CODING SECTION

```
 7332                           ;*****************************************************************************************
*
 7333                           ;
 7334                           ;               FVTA.SWQ
 7335                           ;
 7336                           ;*****************************************************************************************
*
 7338
 7339
 7340                           .SBTTL  SOFTWARE PARAMETER CODING SECTION
 7341
 7342                           ;++
 7343                           ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
 7344                           ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 7345                           ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 7346                           ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 7347                           ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 7348                           ; WITH THE OPERATOR.
 7349                           ;--
 7350
 7351 037052                           BGNSFT
      037052  000013                                                        .WORD L10062-L$SOFT/2
      037054                                                          L$SOFT::
 7352
 7361
 7362 037054                   ;UNIT NUMBER PRINTOUT QUESTION:
                                       GPRML   SWPTQ1,0,20,YES
      037054  000130                                                        .WORD   T$CODE
      037056  037102                                                        .WORD   SWPTQ1
      037060  000020                                                        .WORD   20
 7363                          ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
 7364 037062                           GPRMD   SWPTQ2,2,D,177777,0,177777,YES
      037062  001052                                                        .WORD   T$CODE
      037064  037156                                                        .WORD   SWPTQ2
      037066  177777                                                        .WORD   177777
      037070  000000                                                        .WORD   T$LOLIM
      037072  177777                                                        .WORD   T$HILIM
 7365                          ;ROM VERSION PRINTOUT ON FIRST PASS QUESTION:
 7366 037074                           GPRML   SWPTQ3,0,1,YES
      037074  000130                                                        .WORD   T$CODE
      037076  037245                                                        .WORD   SWPTQ3
      037100  000001                                                        .WORD   1
 7367
 7368                                  .EVEN
 7369
 7370 037102                           ENDSFT
                                                                           .EVEN
      037102                                                          L10062:
 7371
 7372
 7379 037102    122     105     120    SWPTQ1: .ASCIZ  /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /
      037105    117     122     124
      037110    040     125     116
      037113    111     124     040
      037116    116     125     115
      037121    102     105     122
      037124    040     101     123
      037127    040     105     101
      037132    103     110     040
      037135    125     116     111
```

SOFTWARE PARAMETER CODING SECTION

```
                037140  124  040  111
                037143  123  040  124
                037146  105  123  124
                037151  105  104  072
                037154  040  000
7380            037156  116  125  115        SWPTQ2: .ASCIZ  /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /
                037161  102  105  122
                037164  040  117  106
                037167  040  111  116
                037172  104  111  126
                037175  111  104  125
                037200  101  114  040
                037203  104  101  124
                037206  101  040  105
                037211  122  122  117
                037214  122  123  040
                037217  124  117  040
                037222  122  105  120
                037225  117  122  124
                037230  040  117  116
                037233  040  101  040
                037236  114  111  116
                037241  105  072  040
                037244  000
7381            037245  122  117  115        SWPTQ3: .ASCIZ  /ROM VERSION PRINTOUT ON THE FIRST PASS: /
                037250  040  126  105
                037253  122  123  111
                037256  117  116  040
                037261  120  122  111
                037264  116  124  117
                037267  125  124  040
                037272  117  116  040
                037275  124  110  105
                037300  040  106  111
                037303  122  123  124
                037306  040  120  101
                037311  123  123  072
                037314  040  000
7382                                        .EVEN
```

SOFTWARE PARAMETER CODING SECTION

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


037316                  $PATCH::
037316                          .BLKW   24



                                LASTAD
                                                                        .EVEN
037366  000000                                                          .WORD   0
037370  000000                                                          .WORD   0
037372                  L$LAST::
037372                          ENDMOD



                        .END
```

SYMBOL TABLE

```
ACTLNS  002236 G    CTRLCF  002352 G    DRADRT  002244 G    EM0603  010305 G    EM3005  015054 G
ADR     000020 G    C$AU    000052      DROP    025144      EM0701  010470 G    EM3101  015146 G
ADRPTR  020136 G    C$AUTO  000061      DR00MG  006002 G    EM0702  010525 G    EM3102  015202 G
ALTFLD  017626 G    C$BRK   000022      DR02MG  006006 G    EM0703  010705 G    EM9009  015243 G
ASSEMB  000010      C$BSEG  000004      DR04MG  006013 G    EM0801  011070 G    EM9010  015267 G
BCOUNT  002372 G    C$BSUB  000002      DR06MG  006017 G    EM0802  011130 G    EM9014  015313 G
BITTBL  002410 G    C$CEFG  000045      DR10MG  006024 G    EM0901  011203 G    EM9017  015407 G
BIT0    000001 G    C$CLCK  000062      DR12MG  006033 G    EM0902  011234 G    EM9018  015520 G
BIT00   000001 G    C$CLEA  000012      DR14MG  006044 G    EM1001  011270 G    EM9019  015530 G
BIT01   000002 G    C$CLOS  000035      DR16MG  006055 G    EM1002  011334 G    EM9020  015545 G
BIT02   000004 G    C$CLP1  000006      EDROP   025222      EM1003  011421 G    EM9022  015571 G
BIT03   000010 G    C$CVEC  000036      EF.CON  000036 G    EM1101  011477 G    EM9023  015610 G
BIT04   000020 G    C$DCLN  000044      EF.NEW  000035 G    EM1201  011523 G    EM9024  015632 G
BIT05   000040 G    C$DODU  000051      EF.PWR  000034 G    EM1202  011542 G    EM9026  015650 G
BIT06   000100 G    C$DRPT  000024      EF.RES  000037 G    EM1203  011626 G    EM9301  015674 G
BIT07   000200 G    C$DU    000053      EF.STA  000040 G    EM1204  011704 G    EM9302  015714 G
BIT08   000400 G    C$EDIT  000003      EF0503  004132 G    EM1205  011740 G    EM9303  015744 G
BIT09   001000 G    C$ERDF  000055      EF0505  004137 G    EM1301  011764 G    EM9304  016011 G
BIT1    000002 G    C$ERHR  000056      EF1401  004212 G    EM1302  012010 G    ENDETB  003726 G
BIT10   002000 G    C$ERRO  000060      EF1402  004314 G    EM1401  012047 G    ENDIT   025064
BIT11   004000 G    C$ERSF  000054      EF1601  004351 G    EM1402  012077 G    ERCNTB  002464 G
BIT12   010000 G    C$ERSO  000057      EF1602  004375 G    EM1403  012165 G    ERLTBL  002726 G
BIT13   020000 G    C$ESCA  000010      EF1603  004437 G    EM1404  012240 G    ERRBLK  003774 G
BIT14   040000 G    C$ESEG  000005      EF1604  004501 G    EM1405  012312 G    ERRMSG  003772 G
BIT15   100000 G    C$ESUB  000003      EF3001  004576 G    EM1406  012325 G    ERRNBR  003770 G
BIT2    000004 G    C$ETST  000001      EF3002  004645 G    EM1407  012340 G    ERRTYP  003766 G
BIT3    000010 G    C$EXIT  000032      EF9001  004714 G    EM1408  012352 G    ERSMRF  002462 G
BIT4    000020 G    C$GETB  000026      EF9002  004776 G    EM1601  012360 G    ER0101  016066 G
BIT5    000040 G    C$GETW  000027      EF9003  005055 G    EM1604  012443 G    ER0201  016404 G
BIT6    000100 G    C$GMAN  000043      EF9004  005111 G    EM1701  012511 G    ER0503  016466 G
BIT7    000200 G    C$GPHR  000042      EF9005  005146 G    EM1801  012566 G    ER0504  016512 G
BIT8    000400 G    C$GPLO  000030      EF9006  005177 G    EM1901  012634 G    ER1401  016560 G
BIT9    001000 G    C$GPRI  000040      EF9010  005223 G    EM2001  012711 G    ER1601  016710 G
BMPCQB  002526 G    C$INIT  000011      EF9016  005322 G    EM2002  012753 G    ER1603  017006 G
BMPCQE  002726 G    C$INLP  000020      EF9017  005417 G    EM2101  013026 G    ER3001  017066 G
BMPCQP  002524 G    C$MANI  000050      EF9018  005473 G    EM2102  013071 G    ER9004  017156 G
BOE     000400 G    C$MEM   000031      EF9019  005600 G    EM2201  013165 G    ER9007  017260 G
BRLEVL  002242 G    C$MSG   000023      EF9301  005624 G    EM2202  013222 G    ER9008  017336 G
BUFBAS  002726 G    C$OPEN  000034      EF9302  005702 G    EM2203  013314 G    ER9101  017420 G
BUFEND  003726 G    C$PNTB  000014      EM0101  020532 G    EM2301  013421 G    ER9301  017446 G
BUFMID  003326 G    C$PNTF  000017      EM0102  020616 G    EM2302  013457 G    EVL     000004 G
BUFPTR  002324 G    C$PNTS  000016      EM0103  006065 G    EM2401  013515 G    E$END   002100
BUF3QT  003526 G    C$PNTX  000015      EM0201  006123 G    EM2601  013551 G    E$LOAD  000035
CACHRX  023634 G    C$QIO   000377      EM0202  006171 G    EM2602  013601 G    F$AU    000015
CACHTX  023662 G    C$RDBU  000007      EM0203  006344 G    EM2603  013672 G    F$AUTO  000020
CALMSL  017700 G    C$REFG  000047      EM0204  006507 G    EM2604  013760 G    F$BGN   000040
CKTRAP  020124 G    C$RESE  000033      EM0301  006666 G    EM2605  014054 G    F$CLEA  000007
CLKBRL  002356 G    C$REVI  000003      EM0302  006731 G    EM2606  014137 G    F$DU    000016
CLKCSR  002354 G    C$RFLA  000021      EM0303  007071 G    EM2607  014176 G    F$END   000041
CLKHRZ  002362 G    C$RPT   000025      EM0401  007250 G    EM2608  014272 G    F$HARD  000004
CLKINT  023710 G    C$SEFG  000046      EM0402  007317 G    EM2609  014343 G    F$HW    000013
CLKVEC  002360 G    C$SPRI  000041      EM0501  007467 G    EM2610  014422 G    F$INIT  000006
CLNRST  020154 G    C$SVEC  000037      EM0502  007535 G    EM2611  014514 G    F$JMP   000050
CLR16W  020176 G    C$TPRI  000013      EM0525  007711 G    EM3001  014577 G    F$MOD   000000
CNTERR  020220 G    DELAY   020276 G    EM0526  010001 G    EM3002  014630 G    F$MSG   000011
CSRA    002244 G    DFPTBL  002214 G    EM0601  010071 G    EM3003  014724 G    F$PROT  000021
CSR0    000000 G    DIAGMC  000000      EM0602  010125 G    EM3004  015000 G    F$PWR   000017
```

SYMBOL TABLE

```
F$RPT = 000012          I$TST = 000041          L$SPTP  002024 G      MFUNIT  004074 G      RXIE1   022242 G
F$SEG = 000003          J$JMP = 000167          L$STA   002030 G      MMENAB  002404 G      RXINPT  024050 G
F$SOFT= 000005          LNCTRA  002254 G        L$SW    002226 G      MMPRES  002402 G      RXINTC  002334 G
F$SRV = 000010          LNCTRO= 000010 G        L$TEST  002114 G      MMSRO   002400 G      RXINTF  002336 G
F$SUB = 000002          LOE   = 040000 G        L$TIML  002014 G      MSG1    016172 G      RXVECA  002232 G
F$SW  = 000014          LOT   = 000010 G        L$UNIT  002012 G      MSG2    016250 G      ROSLOT= 000002 G
F$TEST= 000001          LPCSLT= 000036 G        L10000  002224        MSG3    016327 G      R1SLOT= 000004 G
GETPRM  024650          LPRA    002250 G        L10001  002232        MSLCNT  002376 G      R2SLOT= 000006 G
GPRSOB  002450 G        LPRO  = 000004 G        L10002  016170        MSLGET  020336 G      R3SLOT= 000010 G
G$CNTO= 000200          L$ACP   002110 G        L10003  016464        MSLOOP  020452 G      R4SLOT= 000012 G
G$DELM= 000372          L$APT   002036 G        L10004  016510        MSTICK  002374 G      R5SLOT= 000014 G
G$DISP= 000003          L$AU    025230 G        L10005  016556        NDERPT  002230 G      SAVBMP  022266 G
G$EXCP= 000400          L$AUT   002070 G        L10006  016706        NEWPAS  024630        SFPTBL  002226 G
G$HILI= 000002          L$AUTO  025100 G        L10007  017004        NEWRES  024622        SKPSTS  022334 G
G$LOLI= 000001          L$CCP   002106 G        L10010  017064        NEWSTA  024312        STATA   002252 G
G$NO  = 000000          L$CLEA  025102 G        L10011  017154        NUMLNS= 000010 G      STATO = 000006 G
G$OFFS= 000400          L$CO    002032 G        L10012  017256        OOPS    020466 G      SVCGBL= 000000
G$OFSI= 000376          L$DEPO  002011 G        L10013  017334        OPTION  002226 G      SVCINS= 000001
G$PRMA= 000001          L$DESC  004046 G        L10014  017416        O$APTS= 000000        SVCSUB= 000001
G$PRMD= 000002          L$DESP  002076 G        L10015  017444        O$AU  = 000000        SVCTAG= 000001
G$PRML= 000000          L$DEVP  002060 G        L10016  017624        O$BGNR= 000001        SVCTST= 000001
G$RADA= 000140          L$DISP  002124 G        L10017  024236        O$BGNS= 000001        SWAPO   022412 G
G$RADB= 000000          L$DLY   002116 G        L10021  025076        O$DU  = 000001        SWPTQ1  037102
G$RADD= 000040          L$DTP   002040 G        L10022  025100        O$ERRT= 000001        SWPTQ2  037156
G$RADL= 000120          L$DTYP  002034 G        L10023  025116        O$GNSW= 000001        SWPTQ3  037245
G$RADO= 000020          L$DU    025120 G        L10024  025226        O$POIN= 000001        S$LSYM= 010000
G$XFER= 000004          L$DUT   002072 G        L10025  025234        O$SETU= 000000        TIMER1  002364 G
G$YES = 000010          L$DVTY  004036 G        L10026  025524        PAROA   002406 G      TIMER2  002366 G
HELP  = 000000          L$EF    002052 G        L10027  025754        PASCNT  002332 G      TIMER3  002370 G
HOE   = 100000 G        L$ENVI  002044 G        L10030  026220        PCSLOT= 000016 G      TNUM  = 000033 G
HWPTQ1  036726          L$ERRT  003766 G        L10031  026416        PNT   = 001000 G      TP4FLG  002346 G
HWPTQ2  036744          L$ETP   002102 G        L10032  026610        PREGRT  004020 G      TP4RTN  024134 G
HWPTQ3  036777          L$EXP1  002046 G        L10033  027026        PREG05  003776        TP4VEC  002344 G
HWPTQ5  037025          L$EXP4  002064 G        L10034  027234        PRI   = 002000 G      TSABRT  022462 G
IBE   = 010000 G        L$EXP5  002066 G        L10035  027442        PRI00 = 000000 G      TSTNUM  002326 G
IDU   = 000040 G        L$HARD  036662 G        L10036  027630        PRI01 = 000040 G      TXAD1A  002256 G
IER   = 020000 G        L$HIME  002120 G        L10037  030044        PRI02 = 000100 G      TXAD1O= 000012 G
IESTAT  002330 G        L$HPCP  002016 G        L10040  030272        PRI03 = 000140 G      TXAD2A  002260 G
ISR   = 000100 G        L$HPTP  002022 G        L10041  030554        PRI04 = 000200 G      TXAD2O= 000014 G
IXE   = 004000 G        L$HW    002214 G        L10042  031132        PRI05 = 000240 G      TXBFCA  002262 G
I$AU  = 000041          L$ICP   002104 G        L10043  031532        PRI06 = 000300 G      TXBFCO= 000016 G
I$AUTO= 000041          L$INIT  024246 G        L10044  031750        PRI07 = 000340 G      TXCHA   002246 G
I$CLN = 000041          L$LADP  002026 G        L10045  032172        PUFIFO  020714 G      TXCHRO= 000002 G
I$DU  = 000041          L$LAST  037372 G        L10046  032444        RBUFA   002246 G      TXDSBL  022574 G
I$HRD = 000041          L$LOAD  002100 G        L10047  032722        RBUFO = 000002 G      TXENBL  022670 G
I$INIT= 000041          L$LUN   002074 G        L10050  033036        RDPDR   020776 G      TXIEO   022764 G
I$MOD = 000041          L$MREV  002050 G        L10051  033254        REGTST  021216 G      TXIE1   023024 G
I$MSG = 000041          L$NAME  002000 G        L10052  033516        REPSMR  021426 G      TXINTC  002340 G
I$PROT= 000040          L$PRIO  002042 G        L10053  034024        RESETT  021454 G      TXINTF  002342 G
I$PTAB= 000041          L$PROT  024240 G        L10054  034370        RMATBB  002304 G      TXINTR  024156 G
I$PWR = 000041          L$PRT   002112 G        L10055  035422        ROLDAP  021566 G      TXVECA  002234 G
I$RPT = 000041          L$REPP  002062 G        L10056  036312        RSTRPT  021620 G      T$ARGC= 000003
I$SEG = 000041          L$REV   002010 G        L10057  036576        RXBDTX= 000030 G      T$CODE= 000130
I$SETU= 000041          L$RPT   024232 G        L10060  036656        RXBETX= 000020 G      T$ERRN= 022125
I$SFT = 000041          L$SOFT  037054 G        L10061  036726        RXBFUL= 000100 G      T$EXCP= 000000
I$SRV = 000041          L$SPC   002056 G        L10062  037102        RXBRRT  023760 G      T$FLAG= 000050
I$SUB = 000041          L$SPCP  002020 G        MAPLNS= 000377 G      RXIEO   022202 G      T$GMAN= 000000
```

SYMBOL TABLE

```
T$HILI= 177777          T$TEST= 000033          T$$TES= 010060          T22     033520 G        UNITN   002240 G
T$LAST= 000001          T$TSTM= 177777          T1      025236 G        T23     034026 G        UNSDIV  023050 G
T$LOLI= 000000          T$TSTS= 000001          T10     027632 G        T24     034372 G        WAIBIC  023204 G
T$LSYM= 010000          T$$AU = 010025          T11     030046 G        T25     035424 G        WAIBIS  023260 G
T$LTNO= 000033          T$$AUT= 010022          T12     030274 G        T26     036314 G        WDPDR   023334 G
T$NEST= 177777          T$$CLE= 010023          T13     030556 G        T27     036600 G        WORD1   002350 G
T$NSO = 000000          T$$DU = 010024          T14     031134 G        T3      025756 G        WTWLNC  023530 G
T$NS1 = 000005          T$$HAR= 010061          T15     031534 G        T4      026222 G        WTWLNS  023560 G
T$PTNU= 000000          T$$HW = 010000          T16     031752 G        T5      026420 G        WTWLPR  023604 G
T$SAVL= 177777          T$$INI= 010021          T17     032174 G        T6      026612 G        X$ALWA= 000000
T$SEGL= 177777          T$$MSG= 010016          T18     032446 G        T7      027030 G        X$FALS= 000040
T$SUBN= 000000          T$$PRO= 010020          T19     032724 G        T8      027236 G        X$OFFS= 000400
T$TAGL= 177777          T$$RPT= 010017          T2      025526 G        T9      027444 G        X$TRUE= 000020
T$TAGN= 010063          T$$SOF= 010062          T20     033040 G        UAM   = 000200 G        $PATCH  037316 G
T$TEMP= 000000          T$$SW = 010001          T21     033256 G        UNBTTB  002264 G

. ABS.  037372      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  28671 WORDS  ( 112 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:04:36
VAAOEO.OBJ,VAAOEO.LST/-SP=SVC34R/ML,VAAOEO.P11
```