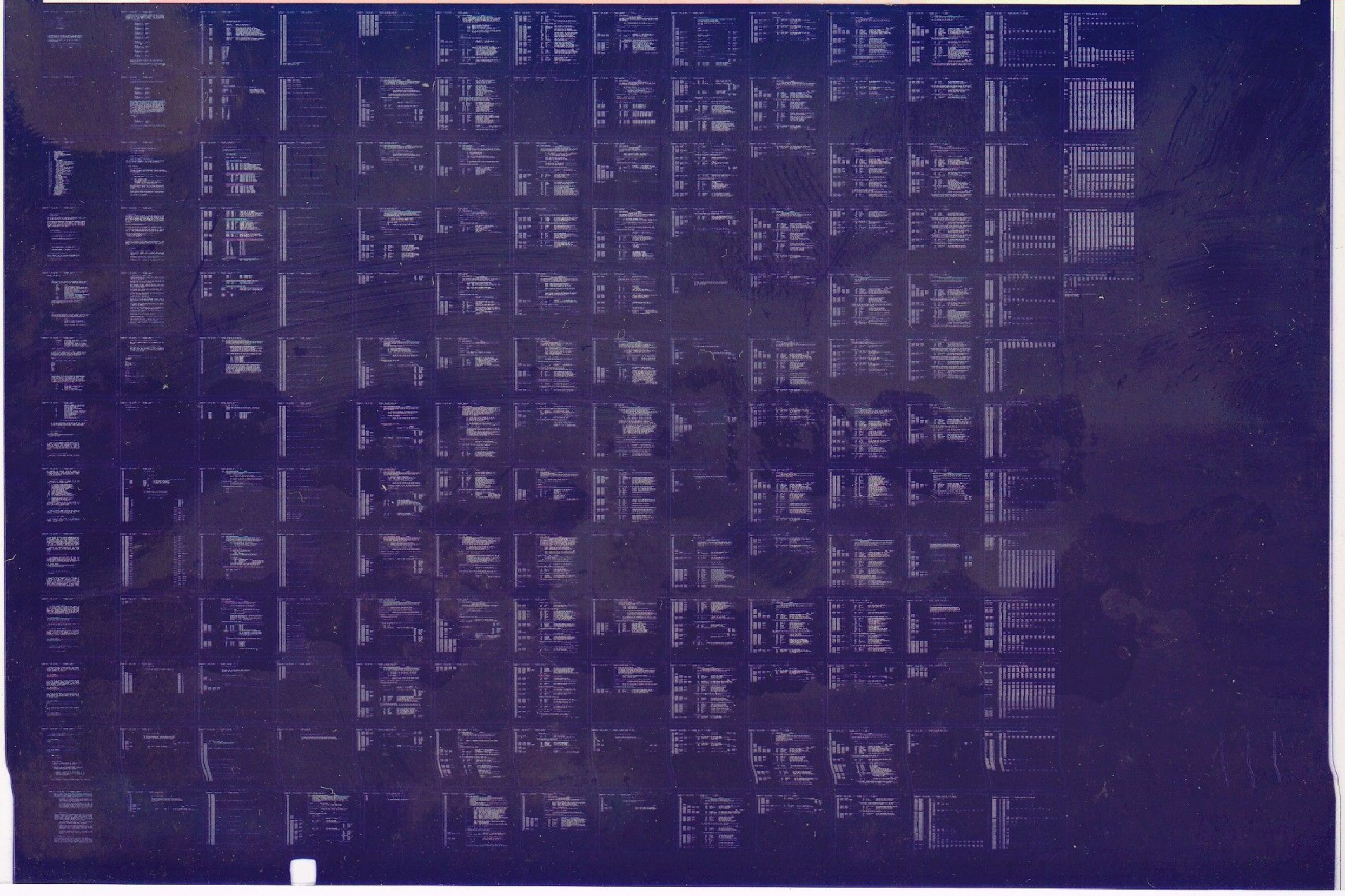
DHV11

DHV-11 FUNC TST PT1
CVDHAA0

AH-T653A-MC
FICHE 1 OF 1



```
.REM &
```

IDENTIFICATION

PRODUCT CODE: AC-T652A-MC

PRODUCT NAME: CVDHAA0 DHV-11 FUNC TST PART1

PRODUCT DATE: 31 OCTOBER 1983

MAINTAINER: EDSHE - DIAGNOSTICS GROUP

AUTHOR: BERT KLEINSCHMIDT
TONY GRIMSHAW

***** MODIFICATION HISTORY *****

ORIGINAL RELEASE: 31-OCT-83 BERT KLEINSCHMIDT

TABLE OF CONTENTS

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 RESET AND REGISTER ACCESS FUNCTIONS OF THE DHV BOARD UNDER TEST 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 (NOT RELATED TO BELL PROMPTING).
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION (ILLEGAL FOR THIS DIAGNOSTIC).
ISR	INHIBIT STATISTICAL REPORTS.
IDU	INHIBIT DROPPING OF UNITS BY DIAGNOSTIC. (HAS NO EFFECT IN THIS 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 (160020 OCTAL).

2. 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. THE DEFAULT ANSWER FOR THIS QUESTION IS 377 OCTAL WHICH SELECTS ALL LINES.

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

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

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

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

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

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

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

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

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

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

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

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

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

2.8 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

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

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

THE GENERAL ERROR MESSAGE IS OF THE FORM:

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

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

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

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

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

MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 ERROR MESSAGES

THIS PROGRAM IS INTENDED TO PROVIDE A GO/NO-GO INDICATION OF THE FUNCTIONALITY OF 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 (O).
    ACTUAL DATA:   000023 (O).
```

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. RECEPTION HANDSHAKING TESTS - THESE TESTS VERIFY THAT THE HANDSHAKING BITS WHICH ARE NECESSARY FOR THE READING OF CODES FROM THE FIFO ARE WORKING.

5. 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.

6. 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.

7. 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.

8. 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.

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

10. 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.

11. 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.

12. 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.

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

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

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

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

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

18. BYTE ACCESS READ/WRITE TEST - VERIFIES THAT THE REGISTERS

RESPOND CORRECTLY TO BYTE READ AND WRITE ACCESSES.

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

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

21. 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 CVDHAA0
CVDHAA0.BIC

DRS
CVDHAA0
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)  160020 ? ^Z

UNIT 1
CSR ADDRESS:  (O)  160020 ? 160040
ACTIVE LINE BIT MAP:  (O)  377 ? 10

CHANGE SW (L) ? N

TESTING UNIT :  0

 ROM VERSION NUMBERS: PROC_1 =  1   PROC_2 =  1
```

```
        TESTING UNIT

         ROM VERSION NUMBERS: PROC_1 =  1   PROC_2 =  1

        CVDHA EOP    1
            0 CUMULATIVE ERRORS

        ...

                &
```

```
1004                                    .LIST SEQ,LOC,BIN,MEB
1005
1006                                    .NLIST CND
1007
1008
1009                                    .SBTTL  PROGRAM HEADER
1010
1011
1012                                            .MCALL  SVC
1013  000000'                                   SVC                     ; INITIALIZE SUPERVISOR MACROS
1014          000001                    SVCINS= 1       ; LIST INSTRUCTIONS, SHIFTED RIGHT
1015          000001                    SVCTST= 1       ; LIST TEST TAGS, SHIFTED RIGHT
1016          000001                    SVCSUB= 1       ; LIST SUBTEST TAGS, SHIFTED RIGHT
1017          000001                    SVCGBL= 1       ; LIST GLOBAL TAGS, SHIFTED RIGHT
1018          000001                    SVCTAG= 1       ; LIST OTHER TAGS, SHIFTED RIGHT
1019                                    .ENABL  ABS
1020                                    .ENABL  AMA
1021          002000                    .       =       2000
1022
1023  002000                                    BGNMOD
1024
1025                                    ;++
1026                                    ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1027                                    ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1028                                    ;--
1029
1030  002000                                    POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL
1031
1032
1033  002000                            HEADER  CVDHA,A,0,15,0,PRI07
1034  002000
1035  002000       103                  L$NAME::
1036  002001       126                          .ASCII /C/
1037  002002       104                          .ASCII /V/
1038  002003       110                          .ASCII /D/
1039  002004       101                          .ASCII /H/
1040  002005       000                          .ASCII /A/
1041  002006       000                          .BYTE   0
1042  002007       000                          .BYTE   0
1043  002010                                    .BYTE   0
1044  002010       101                  L$REV::
1045  002011                                    .ASCII  /A/
1046  002011       060                  L$DEPO::
1047  002012                                    .ASCII  /0/
1048  002012  000000                    L$UNIT::
1049  002014                                    .WORD   0
1050  002014  000015                    L$TIML::
1051  002016                                    .WORD   15
1052  002016  025612                    L$HPCP::
1053  002020                                    .WORD   L$HARD
1054  002020  025702                    L$SPCP::
1055  002022                                    .WORD   L$SOFT
1056  002022  002176                    L$HPTP::
1057  002024                                    .WORD   L$HW
1058  002024  002204                    L$SPTP::
1059  002026                                    .WORD   L$SW
                                        L$LADP::
```

```
1060  002026  026220
1061  002030
1062  002030  000000
1063  002032
1064  002032  000000
1065  002034
1066  002034  000000
1067  002036
1068  002036  000000
1069  002040
1070  002040  002124
1071  002042
1072  002042  000340
1073  002044
1074  002044  000000
1075  002046
1076  002046  000000
1077  002050
1078  002050      003
1079  002051      003
1080  002052
1081  002052  000000
1082  002054  000000
1083  002056
1084  002056  000000
1085  002060
1086  002060  002734
1087  002062
1088  002062  017036
1089  002064
1090  002064  000000
1091  002066
1092  002066  000000
1093  002070
1094  002070  000000
1095  002072
1096  002072  017666
1097  002074
1098  002074  000000
1099  002076
1100  002076  002744
1101  002100
1102  002100  104035
1103  002102
1104  002102  002664
1105  002104
1106  002104  017052
1107  002106
1108  002106  017650
1109  002110
1110  002110  017646
1111  002112
1112  002112  017044
1113  002114
1114  002114  000000
1115  002116
```

```
                .WORD   L$LAST
L$STA::
                .WORD   0
L$CO::
                .WORD   0
L$DTYP::
                .WORD   0
L$APT::
                .WORD   0
L$DTP::
                .WORD   L$DISPATCH
L$PRIO::
                .WORD   PRI07
L$ENVI::
                .WORD   0
L$EXP1::
                .WORD   0
L$MREV::
                .BYTE   C$REVISION
                .BYTE   C$EDIT
L$EF::
                .WORD   0
                .WORD   0
L$SPC::
                .WORD   0
L$DEVP::
                .WORD   L$DVTYP
L$REPP::
                .WORD   L$RPT
L$EXP4::
                .WORD   0
L$EXP5::
                .WORD   0
L$AUT::
                .WORD   0
L$DUT::
                .WORD   L$DU
L$LUN::
                .WORD   0
L$DESP::
                .WORD   L$DESC
L$LOAD::
                EMT     E$LOAD
L$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::
```

```
  1116  002116  000000                                          .WORD   0
  1117  002120                                        L$HIME::
  1118  002120  000000                                          .WORD   0
  1119
```

```
1120                                    .SBTTL  DISPATCH TABLE
1121
1122                                    ;++
1123                                    ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1124                                    ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1125                                    ;--
1126
1127  002122                                    DISPATCH 20
1128  002122  000024                                                    .WORD   20
1129  002124                                                  L$DISPATCH::
1130  002124  020004                                                    .WORD   T1
1131  002126  020274                                                    .WORD   T2
1132  002130  020524                                                    .WORD   T3
1133  002132  020770                                                    .WORD   T4
1134  002134  021166                                                    .WORD   T5
1135  002136  021360                                                    .WORD   T6
1136  002140  021576                                                    .WORD   T7
1137  002142  022004                                                    .WORD   T8
1138  002144  022212                                                    .WORD   T9
1139  002146  022400                                                    .WORD   T10
1140  002150  022614                                                    .WORD   T11
1141  002152  023042                                                    .WORD   T12
1142  002154  023324                                                    .WORD   T13
1143  002156  023624                                                    .WORD   T14
1144  002160  024224                                                    .WORD   T15
1145  002162  024442                                                    .WORD   T16
1146  002164  024664                                                    .WORD   T17
1147  002166  025136                                                    .WORD   T18
1148  002170  025414                                                    .WORD   T19
1149  002172  025530                                                    .WORD   T20
1150
```

```
1151
1152
1153                                    .SBTTL  DEFAULT HARDWARE P-TABLE
1154
1155                                    ;++
1156                                    ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1157                                    ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
1158                                    ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
1159                                    ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
1160                                    ;--
1161
1162  002174                                    BGNHW   DFPTBL
1163  002174  000002                                                                   .WORD   L10000-L$HW/2
1164  002176                                                                  L$HW::
1165  002176                                                                  DFPTBL::
1166
1167  002176  160020                           .WORD   160020  ;DEFAULT CSR ADDRESS.
1168  002200  000377                           .WORD   377     ;DEFAULT ACTIVE LINES BIT MAP.
1169
1170  002202                                    ENDHW
1171  002202                                                                  L10000:
```

```
1172
1173
1174                                    .SBTTL  SOFTWARE P-TABLE
1175
1176                                    ;++
1177                                    ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
1178                                    ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
1179                                    ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
1180                                    ; AT RUN TIME.
1181                                    ;--
1182
1183  002202                                    BGNSW   SFPTBL
1184  002202  000002                                                                   .WORD   L10001-L$SW/2
1185  002204                                                                  L$SW::
1186  002204                                                                  SFPTBL::
1187
1188  002204  000021                    OPTION::        .WORD   21      ;BIT MAP OF PROGRAM CONTROL FLAGS
1189  002206  000000                    NDERPT::        .WORD   0       ;DEFAULT NUMBER OF INDIVIDUAL DATA ERRORS TO RPT
1190
1191  002210                                    ENDSW
1192  002210                                                                  L10001:
```

```
                                .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.


002210                                  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
                000001          BIT00== 1
                                ;
                001000          BIT9==  BIT09
                000400          BIT8==  BIT08
                000200          BIT7==  BIT07
                000100          BIT6==  BIT06
```

```
1249          000040                  BIT5==  BIT05
1250          000020                  BIT4==  BIT04
1251          000010                  BIT3==  BIT03
1252          000004                  BIT2==  BIT02
1253          000002                  BIT1==  BIT01
1254          000001                  BIT0==  BIT00
1255                                  ;
1256                                  ; EVENT FLAG DEFINITIONS
1257                                  ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
1258                                  ;
1259          000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
1260          000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
1261          000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
1262          000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
1263          000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
1264                                  ;
1265                                  ;
1266                                  ; PRIORITY LEVEL DEFINITIONS
1267                                  ;
1268          000340                  PRI07== 340
1269          000300                  PRI06== 300
1270          000240                  PRI05== 240
1271          000200                  PRI04== 200
1272          000140                  PRI03== 140
1273          000100                  PRI02== 100
1274          000040                  PRI01== 40
1275          000000                  PRI00== 0
1276                                  ;
1277                                  ;OPERATOR FLAG BITS
1278                                  ;
1279              000004              EVL==       4
1280              000010              LOT==      10
1281              000020              ADR==      20
1282              000040              IDU==      40
1283              000100              ISR==     100
1284              000200              UAM==     200
1285              000400              BOE==     400
1286              001000              PNT==    1000
1287              002000              PRI==    2000
1288              004000              IXE==    4000
1289              010000              IBE==   10000
1290              020000              IER==   20000
1291              040000              LOE==   40000
1292              100000              HOE==  100000
1293
```

```
1294
1295
1296                                    .SBTTL  GLOBAL DATA SECTION
1297
1298                                    ;++
1299                                    ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1300                                    ; IN MORE THAN ONE TEST.
1301                                    ;--
1302
1303                                    ;*****************************************************************************
1304                                    ;       UNIT VARIBLE AREA
1305                                    ;*****************************************************************************
1306
1307  002210  000377                            ACTLNS:: .WORD  377     ;ACTIVE LINE BIT MAP.
1308  002212  000000                            UNITN:: .WORD   0       ;UNIT NUMBER.
1309
1310                                    ;*****************************************************************************
1311                                    ;       DEVICE REGISTER ADDRESS TABLE
1312                                    ;*****************************************************************************
1313  002214                                    DRADRT::
1314  002214  160020                            CSRA::  .WORD   160020  ;DHV-11 CSR ADDRESS
1315  002216  160022                    TXCHA:: RBUFA:: .WORD   160022  ;DHV-11 RECEIVE/TRANSMIT BUFFER ADDRESS
1316  002220  160024                            LPRA::  .WORD   160024  ;DHV-11 LINE PARAMETER REGISTER ADDRESS
1317  002222  160026                            STATA:: .WORD   160026  ;DHV-11 STATUS REGISTER ADDRESS
1318  002224  160030                            LNCTRA:: .WORD  160030  ;DHV-11 LINE CONTROL REGISTER ADDRESS
1319  002226  160032                            TXAD1A:: .WORD  160032  ;DHV-11 TRANSMIT BUFFER 1 REGISTER ADDRESS
1320  002230  160034                            TXAD2A:: .WORD  160034  ;DHV-11 TRANSMIT BUFFER 2 REGISTER ADDRESS
1321  002232  160036                            TXBFCA:: .WORD  160036  ;DHV-11 TRANSMIT BUFFER COUNT REGISTER ADDRESS
1322
1323
1324                                    ;*****************************************************************************
1325                                    ;       BIT MASK TABLE OF UN-USED DHV DEVICE REGISTER BITS.
1326                                    ;*****************************************************************************
1327  002234  137660                            UNBTTB::.WORD   137660  ;UNUSED BIT MASK FOR THE CSR
1328  002236  177777                                    .WORD   177777  ;UNUSED BIT MASK FOR THE RBUF/TX REG
1329  002240  000007                                    .WORD   7       ;UNUSED BIT MASK FOR THE LPR
1330  002242  177777                                    .WORD   177777  ;UNUSED BIT MASK FOR THE STAT
1331  002244  166051                                    .WORD   166051  ;UNUSED BIT MASK FOR THE LNCTRL
1332  002246  000000                                    .WORD   0       ;UNUSED BIT MASK FOR THE TBUFFAD1
1333  002250  077700                                    .WORD   77700   ;UNUSED BIT MASK FOR THE TBUFFAD2
1334  002252  000000                                    .WORD   0       ;UNUSED BIT MASK FOR THE TBUFFCT
1335
1336                                    ;*****************************************************************************
1337                                    ;       REGISTER MESSAGE ADDRESS TABLE
1338                                    ;*****************************************************************************
1339  002254  004507                            RMATBB:: .WORD  DR00MG  ;ADDRESS OF "CSR" MESSAGE.
1340  002256  004513                                    .WORD   DR02MG  ;ADDRESS OF "RBUF" MESSAGE.
1341  002260  004520                                    .WORD   DR04MG  ;ADDRESS OF "LPR" MESSAGE.
1342  002262  004524                                    .WORD   DR06MG  ;ADDRESS OF "STAT" MESSAGE.
1343  002264  004531                                    .WORD   DR10MG  ;ADDRESS OF "LNCTRL" MESSAGE.
1344  002266  004540                                    .WORD   DR12MG  ;ADDRESS OF "TBUFFAD1" MESSAGE.
1345  002270  004551                                    .WORD   DR14MG  ;ADDRESS OF "TBUFAD2" MESSAGE.
1346  002272  004562                                    .WORD   DR16MG  ;ADDRESS OF "TBUFFCT" MESSAGE.
1347
1348
1349                                    ;*****************************************************************************
                                        ;       ASSORTED GLOBAL VARIABLES:
```

```
1350
1351  002274  000001                ;*****************************************************************************
                                    TSTNUM:: .WORD  1       ;STORAGE FOR THE TEST NUMBER.
1352  002276  000000                IESTAT:: .WORD  0       ;STORAGE FOR THE INTERRUPT ENABLE BIT STATES.
1353  002300  000000                PASCNT:: .WORD  0       ;STO'G FOR PASS COUNT USED IN ROM VERSION# TST.
1354  002302  000000                TP4VEC:: .WORD  0       ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
1355  002304  000000                TP4FLG:: .WORD  0       ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
1356  002306  000000                WORD1::  .WORD  0       ;LOCATION FOR PASSING INDIRECT PARAMETERS.
1357  002310  000000                CTRLCF:: .WORD  0       ;STORAGE FOR THE CONTROL-C FLAG.
1358
1359                                ;*****************************************************************************
1360                                ;       LINE TIME CLOCK VARIABLES AND STORAGE.
1361                                ;*****************************************************************************
1362  002312  177546                CLKCSR:: .WORD  177546  ;CSR ADDRESS OF THE LTC.
1363  002314  000300                CLKBRL:: .WORD  PR106   ;INTERRUPT PRIORITY LEVEL OF THE LTC.
1364  002316  000100                CLKVEC:: .WORD  100     ;INTERRUPT VECTOR ADDRESS OF THE LTC.
1365  002320  000074                CLKHRZ:: .WORD  60.     ;INTERRUPT FREQUENCY OF THE LTC.
1366  002322  000000                TIMER1:: .WORD  0       ;HARDWARE CLOCK COUNTER #1.
1367  002324  000000                TIMER2:: .WORD  0       ;HARDWARE CLOCK COUNTER #2.
1368  002326  000170                TIMER3:: .WORD  120.    ;HARDWARE BREAK COUNTER LOCATION.
1369  002330  000170                BCOUNT:: .WORD  120.    ;BREAK COUNT VALUE IN CLOCK TICKS.
1370  002332  000021                MSTICK:: .WORD  17.     ;NUMBER OF MILLI-SECONDS PER LTC TICK.
1371  002334  000062                MSLCNT:: .WORD  62      ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.
1372
1373                                ;*****************************************************************************
1374                                ;       MEMMORY MANAGEMENT VARIABLES AND FLAGS.
1375                                ;*****************************************************************************
1376  002336  177572                MMSR0:: .WORD   177572  ;ADDRESS OF MEM MGT STATUS REGISTER #0.
1377  002340  000000                MMPRES:: .WORD  0       ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
1378  002342  000000                MMENAB:: .WORD  0       ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).
1379  002344  172340                PAR0A:: .WORD   172340  ;ADDRESS OF MEM MGT PAR #0.
1380
1381                                ;*****************************************************************************
1382                                ;       TABLE OF WORDS WITH CORRESPONDING BIT SET FOR GENERATION OF BIT MAPS.
1383                                ;*****************************************************************************
1384  002346  000001                BITTBL:: .WORD  1               ;BIT 0 SET.
1385  002350  000002                        .WORD   2               ;BIT 1 SET.
1386  002352  000004                        .WORD   4               ;BIT 2 SET.
1387  002354  000010                        .WORD   10              ;BIT 3 SET.
1388  002356  000020                        .WORD   20              ;BIT 4 SET.
1389  002360  000040                        .WORD   40              ;BIT 5 SET.
1390  002362  000100                        .WORD   100             ;BIT 6 SET.
1391  002364  000200                        .WORD   200             ;BIT 7 SET.
1392  002366  000400                        .WORD   400             ;BIT 8 SET.
1393  002370  001000                        .WORD   1000            ;BIT 9 SET.
1394  002372  002000                        .WORD   2000            ;BIT 10 SET.
1395  002374  004000                        .WORD   4000            ;BIT 11 SET.
1396  002376  010000                        .WORD   10000           ;BIT 12 SET.
1397  002400  020000                        .WORD   20000           ;BIT 13 SET.
1398  002402  040000                        .WORD   40000           ;BIT 14 SET.
1399  002404  100000                        .WORD   100000          ;BIT 15 SET.
1400
1401                                ;*****************************************************************************
1402                                ;*      GPR SAVE AREA ZERO.
1403                                ;*****************************************************************************
1404  002406                        GPRS0B::                        ;BASE OF GPR SAVE AREA NUMBER ZERO.
1405  002406  000000                        .WORD   0               ;WORD 1, STORAGE FOR R1.
```

```
1406  002410  000000                  .WORD   0               ;WORD 2, STORAGE FOR R2.
1407  002412  000000                  .WORD   0               ;WORD 3, STORAGE FOR R3.
1408  002414  000000                  .WORD   0               ;WORD 4, STORAGE FOR R4.
1409  002416  000000                  .WORD   0               ;WORD 5, STORAGE FOR R5.
1410
1411                          ;*******************************************************************************
1412                          ;*      TRANSMISSION AND RECEPTION VARIABLES, POINTERS, AND FLAGS.
1413                          ;*******************************************************************************
1414  002420  000000                  ERSMRF:: .WORD  0       ;ERROR SUMMARY REPORT FLAGS.
1415  002422  000020                  ERCNTB:: .BLKW  16.     ;TABLE OF ERROR COUNTERS.
1416
1417                          ;*******************************************************************************
1418                          ;       STORAGE AREA FOR THE BMP CODE QUEUE.
1419                          ;*******************************************************************************
1420  002462  000000          BMPCQP::        .WORD   0       ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
1421  002464  000100          BMPCQB::        .BLKW   64.     ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
1422  002664                  BMPCQE::                        ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1423
1424  002664                          ERRTBL
1425  002664                                                          L$ERRTBL::
1426  002664  000000          ERRTYP::        .WORD   0
1427  002666  000000          ERRNBR::        .WORD   0
1428  002670  000000          ERRMSG::        .WORD   0
1429  002672  000000          ERRBLK::        .WORD   0
1430
1431                                  .EVEN
```

```
1432                            .SBTTL  GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.
1433                            ;*******************************************************************************
1434                            ;*      THERE ARE 4 ROUTINES AND MACRO DEFINITIONS USED FOR THE HANDLING OF
1435                            ;*      GPR VALUES DURING SUBROUTINE CALLS WITHIN THIS PROGRAM.  THE FOUR
1436                            ;*      ROUTINES/MACRO CALLS HAVE THE FOLLOWING NAMES:
1437                            ;*
1438                            ;*      SAVE - MACRO DEFINITION USED AT THE BEGINNING OF A SUBROUTINE TO
1439                            ;*              SAVE THE GPR CONTENTS FOR LATER RESTORATION.
1440                            ;*      PASS - MACRO DEFINITION USED AT THE END OF A SUBROUTINE TO RESTORE
1441                            ;*              THE PREVIOUSLY SAVED GPR CONTENTS AND TO LEAVE THE CONTENTS
1442                            ;*              OF THE SPECIFIED GPR(S) INTACT (NOT RESTORED).
1443                            ;*      PREG05 - SUBROUTINE WHICH IS CALLED FROM THE SAVE AND PASS MACRO
1444                            ;*              EXPANSIONS WHICH ACTUALLY PERFORMS THE ACTIONS ON THE GPRS.
1445                            ;*
1446                            ;*      DURING A SUBROUTINE WHICH USES THESE GPR SAVE ROUTINES THE VALUES
1447                            ;*      OF THE GPRS ARE STORED ON THE STACK IN THE FOLLOWING STACK FRAME:
1448                            ;*
1449                            ;*              SP    -> RET PC INTO PREG05 ROUTINE.
1450                            ;*              SP+2  -> GPR R0 CONTENTS.
1451                            ;*              SP+4  -> GPR R1 CONTENTS.
1452                            ;*              SP+6  -> GPR R2 CONTENTS.
1453                            ;*              SP+8  -> GPR R3 CONTENTS.
1454                            ;*              SP+10 -> GPR R4 CONTENTS.
1455                            ;*              SP+12 -> GPR R5 CONTENTS.
1456                            ;*              SP+14 -> RET PC INTO CALLER OF SUB'TNE WHICH CALLED PREG05.
1457                            ;*
1458                            ;*      EACH LEVEL OF SUB'TNE CALLING USES 8 WORDS OF STACK OVERHEAD.
1459                            ;*      THE SAVE AND PASS MACROS CAN ALSO BE USED IN "STRAIGHT LINE CODE"
1460                            ;*      TO SAVE AND RESTORE THE GPR VALUES.  IN ANY CASE, AFTER THE
1461                            ;*      ISSUING OF A PASS CALL THE GPRS WILL BE RESTORED TO THE VALUES
1462                            ;*      THEY HAD PRIOR TO THE LAST SAVE CALL (EXCEPT FOR THE EXCEPTED,
1463                            ;*      OR PASSED INTACT, GPRS SPECIFIED AS PARAMETERS TO THE PASS CALL)
1464                            ;*      AND THE SP WILL ALSO BE RESTORED TO ITS CONDITION BEFORE THE LAST
1465                            ;*      SAVE CALL.  THE PROGRAMMER MUST BE SURE THAT THE SP HAS THE SAME
1466                            ;*      VALUE WHEN THE PASS MACRO IS CALLED AS IT HAD IMMEDIATELY AFTER
1467                            ;*      THE SAVE MACRO WAS CALLED.
1468                            ;*******************************************************************************
```

```
1469                                    .SBTTL GPR FRAME ACCESS EQUATES
1470                                            ;+++
1471                                            ;EQUATES THAT ALLOW ACCESS TO THE STACK FRAME.  THESE ARE THE
1472                                            ;OFFSETS INTO THE STACK FOR REGISTERS SAVED DURING THE PREGO5
1473                                            ;ROUTINE.
1474                                            ;---
1475
1476            000036                          LPCSLT==        36      ;OFFSET FOR LAST RETURN PC.
1477            000016                          PCSLOT==        16      ;OFFSET FOR RETURN PC.
1478            000014                          R5SLOT==        14      ;OFFSET FOR R5.
1479            000012                          R4SLOT==        12      ;OFFSET FOR R4.
1480            000010                          R3SLOT==        10      ;OFFSET FOR R3.
1481            000006                          R2SLOT==        6       ;OFFSET FOR R2.
1482            000004                          R1SLOT==        4       ;OFFSET FOR R1.
1483            000002                          R0SLOT==        2       ;OFFSET FOR R0.
```

```
1484                            .SBTTL  GLOBAL MACRO DEFINITION        - SAVE -
1485                            ;*****************************************************************************
1486                            ;*      THIS MACRO IS USED AT THE BEGINNING OF A SUBROUTINE TO SAVE THE
1487                            ;*      CONTENTS OF THE GPRS R0 THRU R5.
1488                            ;*
1489                            ;* INPUTS:      SP - UNCHANGED SINCE SUBROUTINE WAS ENTERED
1490                            ;*              R5SLOT - OFFSET TO STACK SLOT FOR R5 (EQUATED TO 14 OCTAL)
1491                            ;*
1492                            ;* OUTPUTS:     GPR SAVE AREA ON THE STACK IS LOADED WITH THE CONTENTS OF GPRS
1493                            ;*              TOP OF STACK - LOADED WITH THE RETURN ADDRESS INTO PREG05
1494                            ;*
1495                            ;* CALLING SEQUENCE:    SAVE
1496                            ;*
1497                            ;* COMMENTS:    NO ARGUMENTS ARE ALLOWED.
1498                            ;*              THE PASS MACRO SHOULD BE CALLED TO RESTORE THE GPR VALUES.
1499                            ;*
1500                            ;* SUBORDINATE ROUTINES CALLED: PREG05.
1501                            ;*****************************************************************************
1502
1503                                    .MACRO  SAVE
1504                                    .LIST
1505                                            JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
1506                                    .NLIST
1507                                    .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 PREG05 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 PREG05 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 PREG05, 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 PREG05 SUBRT.
1554                                    .NLIST
1555                                    .ENDM   PASS
```

```
                        .SBTTL GLOBAL SUBROUTINE                - PREG05 -
                        ;*****************************************************************************
                        ;*      PRESERVE REGISTERS R0 THROUGH R5 FOR SUBROUTINE CALLS.
                        ;*
                        ;* INPUTS:      THE RETURN ADDRESS BACK INTO THE CALLING ROUTINE MUST BE IN
                        ;*              GPR R5.  (I.E.- MACROS USE "JSR  R5,PREG05".)
                        ;*
                        ;* OUTPUTS:     REGISTERS R0 THROUGH R5 ARE SAVED ON THE STACK.
                        ;*
                        ;*CALLING SEQUENCE:     SAVE            ;MACRO EXPANSION CALLS PREG05.
                        ;*                      [SUBROUTINE CODE]...
                        ;*                      PASS            ;MACRO EXPANSION RECALLS PREG05.
                        ;*
                        ;*COMMENTS:     THIS ROUTINE IS RE-ENTRANT.
                        ;*
                        ;*              PARAMETERS MAY BE PASSED OUT OF A SUBROUTINE BY MODIFYING THE
                        ;*              REGISTER SAVE AREA ON THE STACK.  USE THE PASS GPRN MACRO
                        ;*              TO RETURN GPR VALUES INTACT.
                        ;*              USE THE RNSLOT OFFSETS FROM THE SP TO PASS OTHER PARAMETERS.
                        ;*              [EXAMPLE:       MOV    VALUE,R0SLOT(SP)        ]
                        ;*              MAKE SURE THE SP IS AT ITS ORIGINAL VALUE WHEN YOU DO THIS.
                        ;*
                        ;*SUBORDINATE ROUTINES CALLED:  NONE.
                        ;*****************************************************************************

002674                  PREG05:                 ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
002674  010446                  MOV     R4,-(SP)        ;SAVE R4
002676  010346                  MOV     R3,-(SP)        ;SAVE R3
002700  010246                  MOV     R2,-(SP)        ;SAVE R2
002702  010146                  MOV     R1,-(SP)        ;SAVE R1
002704  010046                  MOV     R0,-(SP)        ;SAVE R0
002706  010546                  MOV     R5,-(SP)        ;PUSH RETURN PC ON TOP OF STACK
002710  016605  000014          MOV     R5SLOT(SP),R5   ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS

002714  004736                  JSR     PC,@(SP)+       ;CALL THE SUBROUTINE AT THE RETURN ADDRESS
                                                        ;FROM THE PREG05 CALL, PUTTING THE PRESENT
                                                        ;PC ON THE STACK AS A RETURN ADDRESS INTO
                                                        ;THIS (PREG05) ROUTINE.

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

002716  012605          PREGRT:: MOV    (SP)+,R5        ;PUT RETURN PC IN R5.
002720  012600                  MOV     (SP)+,R0        ;RESTORE R0.
002722  012601                  MOV     (SP)+,R1        ;RESTORE R1.
002724  012602                  MOV     (SP)+,R2        ;RESTORE R2.
002726  012603                  MOV     (SP)+,R3        ;RESTORE R3.
002730  012604                  MOV     (SP)+,R4        ;RESTORE R4.

002732  000205                  RTS     R5              ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05.
                                                          ;RESTORING R5 IN THE PROCESS.
```

```
1609                                    .SBTTL  GLOBAL TEXT SECTION
1610
1611
1612                                    ;++
1613                                    ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1614                                    ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1615                                    ; MORE THAN ONE TEST.
1616                                    ;--
1617
1618                                    ;
1619                                    ; NAMES OF DEVICES SUPPORTED BY PROGRAM
1620                                    ;
1621  002734                                    DEVTYP  <DHV-11>
1622  002734                                                                  L$DVTYP::
1623  002734  044104  026526  030461                                                  .ASCIZ  /DHV-11/
1624  002742     000
1625          002744                                                                  .EVEN
1626
1627
1628                                    ; TEST DESCRIPTION
1629                                    ;
1630  002744                                    DESCRIPT        <DHV-11 FUNC TST PART1>
1631  002744                                                                  L$DESC::
1632  002744  044104  026526  030461                                                  .ASCIZ  /DHV-11 FUNC TST
1633  002752  043040  047125  020103
1634  002760  051524  020124  040520
1635  002766  052122  000061
1636                                                                                  .EVEN
1637                                    .EVEN
1638
1639
```

```
1640
1641
1642                                    ;
1643                                    ; FORMAT STATEMENTS USED IN PRINT CALLS
1644                                    ;
1645
1646
1647
1648
1649
1650            .NLIST BIN
1651            .SBTTL  GLOBAL MESSAGE AREA
1652                    ; ********* FORMAT STATEMENTS **********
1653   002772   MFUNIT:: .ASCIZ /%N%A TESTING UNIT :%D4%A(D)%N/
1654   003000
1655   003006
1656   003014
1657   003022
1658   003030   EF0503:: .ASCIZ /%T%N/
1659   003035   EF1401:: .ASCIZ /%N%A ROM VERSION NUMBERS: PROC_1 = %D2%A(D)   PROC_2 = %D2%A(D)%N/
1660   003042
1661   003050
1662   003056
1663   003064
1664   003072
1665   003100
1666   003106
1667   003114
1668   003122
1669   003130
1670   003136
1671   003137   EF1402:: .ASCIZ /%T%A ROM VERSION NUMBER %T%N/
1672   003144
1673   003152
1674   003160
1675   003166
1676   003174   EF1601:: .ASCIZ /%A  %T%A ABORTED %N/
1677   003202
1678   003210
1679   003216
1680   003220   EF1602:: .ASCIZ /%A    EXPECTED DATA: %O6%A (O).%N/
1681   003226
1682   003234
1683   003242
1684   003250
1685   003256
1686   003262   EF1603:: .ASCIZ /%A    ACTUAL DATA:   %O6%A (O).%N/
1687   003270
1688   003276
1689   003304
1690   003312
1691   003320
1692   003324   EF1604:: .ASCIZ /%A  BAD BIT(S) IN DEVICE %T%A REGISTER FOR LINE %D2%A (D).%N/
1693   003332
1694   003340
1695   003346
```

```
1696  003354
1697  003362
1698  003370
1699  003376
1700  003404
1701  003412
1702  003420
1703  003421  EF9001:: .ASCIZ /%A  UNEXPECTED %T%A FOUND IN RECEIVE CHAR FIFO:%N/
1704  003426
1705  003434
1706  003442
1707  003450
1708  003456
1709  003464
1710  003472
1711  003500
1712  003503  EF9002:: .ASCIZ /%A    CODE IS ASSOCIATED WITH LINE: %D2%A(D)%N/
1713  003510
1714  003516
1715  003524
1716  003532
1717  003540
1718  003546
1719  003554
1720  003562  EF9003:: .ASCIZ /%A      CODE IS: %O3%A(O)%N/
1721  003570
1722  003576
1723  003604
1724  003612
1725  003616  EF9004:: .ASCIZ /%A    %T%A VALUE: %O3%A(O)%N/
1726  003624
1727  003632
1728  003640
1729  003646
1730  003653  EF9005:: .ASCIZ /%A    %T%A VALUE: NONE%N/
1731  003660
1732  003666
1733  003674
1734  003702
1735  003704  EF9006:: .ASCIZ /%A  %T%A %D2%A(D)%N/
1736  003712
1737  003720
1738  003726
1739  003730  EF9010:: .ASCIZ /%A    NUMBER OF ERRORS DETECTED ON LINE %D2%A(D) IS %D5%A(D)%N/
1740  003736
1741  003744
1742  003752
1743  003760
1744  003766
1745  003774
1746  004002
1747  004010
1748  004016
1749  004024
1750  004027  EF9016:: .ASCIZ /%A  UNEXPECTED %T%A FOR LINE %D2%A(D) IN FIFO AFTER RESET:%N/
1751  004034
```

```
1752  004042
1753  004050
1754  004056
1755  004064
1756  004072
1757  004100
1758  004106
1759  004114
1760  004122
1761  004124  EF9017:: .ASCIZ /%A    %T%A (WITH ERROR FLAGS) IS %O6%A(O)%N/
1762  004132
1763  004140
1764  004146
1765  004154
1766  004162
1767  004170
1768  004176
1769  004200  EF9018:: .ASCII /%A  %T%A IN SELFTEST CODE FIFO SLOT FOR LINE %D2/
1770  004206
1771  004214
1772  004222
1773  004230
1774  004236
1775  004244
1776  004252
1777  004260          .ASCIZ /%A(D) AFTER RESET.%N/
1778  004266
1779  004274
1780  004302
1781  004305  EF9019:: .ASCIZ /%A  %T%A %O6%A(O)%N/
1782  004312
1783  004320
1784  004326
1785  004331  EF9301:: .ASCIZ /%A %T%D2%A(D),  BMP CODE REPORTED :%O3%A(O)%N/
1786  004336
1787  004344
1788  004352
1789  004360
1790  004366
1791  004374
1792  004402
1793  004407  EF9302:: .ASCIZ /%A  OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)%N/
1794  004414
1795  004422
1796  004430
1797  004436
1798  004444
1799  004452
1800  004460
1801  004466
1802  004474
1803  004502
1804                ;********* ERROR MESSAGES *************
1805  004507  DR0OMG:: .ASCIZ /CSR/
1806  004513  DR02MG:: .ASCIZ /RBUF/
1807  004520  DR04MG:: .ASCIZ /LPR/
```

```
1808  004524  DR06MG:: .ASCIZ /STAT/
1809  004531  DR10MG:: .ASCIZ /LNCTRL/
1810  004536
1811  004540  DR12MG:: .ASCIZ /TBUFFAD1/
1812  004546
1813  004551  DR14MG:: .ASCIZ /TBUFFAD2/
1814  004556
1815  004562  DR16MG:: .ASCIZ /TBUFFCT/
1816  004570
1817  004572  EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
1818  004600
1819  004606
1820  004614
1821  004622
1822  004630  EM0201:: .ASCIZ /MASTER RESET (PERFORM SELFTEST) TEST /
1823  004636
1824  004644
1825  004652
1826  004660
1827  004666
1828  004674
1829  004676  EM0202:: .ASCIZ /  MASTER RESET BIT DID NOT CLEAR AFTER BOARD RESET./
1830  004704
1831  004712
1832  004720
1833  004726
1834  004734
1835  004742
1836  004750
1837  004756
1838  004762          .ASCIZ /    WAITED 5 SECONDS.  BIT DEFECTIVE OR FIRMWARE HUNG./
1839  004770
1840  004776
1841  005004
1842  005012
1843  005020
1844  005026
1845  005034
1846  005042
1847  005050
1848  005051  EM0203:: .ASCIZ /  MASTER RESET BIT CLEAR IMMEDIATELY AFTER BOARD RESET./
1849  005056
1850  005064
1851  005072
1852  005100
1853  005106
1854  005114
1855  005122
1856  005130
1857  005136
1858  005141          .ASCIZ /    BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
1859  005146
1860  005154
1861  005162
1862  005170
1863  005176
```

```
1864  005204
1865  005212
1866  005214  EM0204:: .ASCIZ \  MR BIT WENT CLEAR WITHIN 1/2 SECOND OF BOARD RESET.\
1867  005222
1868  005230
1869  005236
1870  005244
1871  005252
1872  005260
1873  005266
1874  005274
1875  005302          .ASCIZ /    BIT DEFECTIVE OR SELFTEST WAS (INCORRECTLY) SKIPPED./
1876  005310
1877  005316
1878  005324
1879  005332
1880  005340
1881  005346
1882  005354
1883  005362
1884  005370
1885  005373  EM0301:: .ASCIZ /MASTER RESET (SKIP SELFTEST) TEST /
1886  005400
1887  005406
1888  005414
1889  005422
1890  005430
1891  005436  EM0302:: .ASCIZ /  MR BIT CLR WITHIN 10 MILISECOND AFTER BOARD RESET./
1892  005444
1893  005452
1894  005460
1895  005466
1896  005474
1897  005502
1898  005510
1899  005516
1900  005523          .ASCIZ /    BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
1901  005530
1902  005536
1903  005544
1904  005552
1905  005560
1906  005566
1907  005574
1908  005576  EM0303:: .ASCIZ \  MR BIT WENT CLEAR 1/5 TO 5 SECONDS AFTER RESET.\
1909  005604
1910  005612
1911  005620
1912  005626
1913  005634
1914  005642
1915  005650
1916  005656
1917  005660          .ASCIZ /    SELFTEST DID NOT GET SKIPPED (SHOULD HAVE BEEN SKIPPED)./
1918  005666
1919  005674
```

```
1920  005702
1921  005710
1922  005716
1923  005724
1924  005732
1925  005740
1926  005746
1927  005754
1928  005755  EM0401:: .ASCIZ /RBUF REGISTER RX CHARACTER FIELD TEST /
1929  005762
1930  005770
1931  005776
1932  006004
1933  006012
1934  006020
1935  006024  EM0402:: .ASCIZ /  IMPROPER CODE FOUND IN RX FIFO AFTER DUT RESET./
1936  006032
1937  006040
1938  006046
1939  006054
1940  006062
1941  006070
1942  006076
1943  006104
1944  006106          .ASCIZ /    EXPECTED: SELFTEST CODE.   ACTUAL: IMPROPER CODE./
1945  006114
1946  006122
1947  006130
1948  006136
1949  006144
1950  006152
1951  006160
1952  006166
1953  006174  EM0501:: .ASCIZ /RBUF REGISTER ERROR FLAGS FIELD TEST /
1954  006202
1955  006210
1956  006216
1957  006224
1958  006232
1959  006240
1960  006242  EM0502:: .ASCIZ /  RX ERROR FLAG(S) FOUND CLEAR ON SELFTEST CODE./
1961  006250
1962  006256
1963  006264
1964  006272
1965  006300
1966  006306
1967  006314
1968  006322
1969  006323          .ASCIZ /    EXPECTED: ALL ERROR FLAGS SET,  ACTUAL: FLAG(S) CLEAR./
1970  006330
1971  006336
1972  006344
1973  006352
1974  006360
1975  006366
```

```
1976  006374
1977  006402
1978  006410
1979  006416  EM0601:: .ASCIZ /CSR RX.DATA.AVAIL BIT TEST /
1980  006424
1981  006432
1982  006440
1983  006446
1984  006452  EM0602:: .ASCIZ /  RX.DATA.AVAIL BIT FOUND CLEAR AFTER RESET COMPLETION./
1985  006460
1986  006466
1987  006474
1988  006502
1989  006510
1990  006516
1991  006524
1992  006532
1993  006540
1994  006542          .ASCIZ /    EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
1995  006550
1996  006556
1997  006564
1998  006572
1999  006600
2000  006606
2001  006614
2002  006622
2003  006630
2004  006632  EM0603:: .ASCIZ /  RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO./
2005  006640
2006  006646
2007  006654
2008  006662
2009  006670
2010  006676
2011  006704
2012  006712
2013  006720
2014  006724          .ASCIZ /    600 CHARS READ FROM FIFO WITHOUT R.D.A BIT CLEARING./
2015  006732
2016  006740
2017  006746
2018  006754
2019  006762
2020  006770
2021  006776
2022  007004
2023  007012
2024  007015  EM0701:: .ASCIZ /RBUF RX.DATA.VALID BIT TEST /
2025  007022
2026  007030
2027  007036
2028  007044
2029  007052  EM0702:: .ASCIZ /  RX.DATA.VALID BIT FOUND CLEAR AFTER RESET COMPLETION./
2030  007060
2031  007066
```

```
2032  007074
2033  007102
2034  007110
2035  007116
2036  007124
2037  007132
2038  007140
2039  007142          .ASCIZ /    EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
2040  007150
2041  007156
2042  007164
2043  007172
2044  007200
2045  007206
2046  007214
2047  007222
2048  007230
2049  007232  EM0703:: .ASCIZ /  RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO./
2050  007240
2051  007246
2052  007254
2053  007262
2054  007270
2055  007276
2056  007304
2057  007312
2058  007320
2059  007324          .ASCIZ /    600 CHARS READ FROM FIFO WITHOUT R.D.V BIT CLEARING./
2060  007332
2061  007340
2062  007346
2063  007354
2064  007362
2065  007370
2066  007376
2067  007404
2068  007412
2069  007415  EM0801:: .ASCIZ /RBUF RX.LINE.NUMBER FIELD TEST /
2070  007422
2071  007430
2072  007436
2073  007444
2074  007452
2075  007455  EM0802:: .ASCIZ /  LINE NUMBER IS WRONG ON A SELFTEST CODE./
2076  007462
2077  007470
2078  007476
2079  007504
2080  007512
2081  007520
2082  007526
2083  007530  EM0901:: .ASCIZ /CHECK FOR BMP_CODES TEST/
2084  007536
2085  007544
2086  007552
2087  007560
```

```
 2088  007561  EM0902:: .ASCIZ /UNEXPECTED BMP CODES FOUND./
 2089  007566
 2090  007574
 2091  007602
 2092  007610
 2093  007615  EM1001:: .ASCIZ /DIAGNOSTIC FAIL (SKP SELFTEST) TEST/
 2094  007622
 2095  007630
 2096  007636
 2097  007644
 2098  007652
 2099  007660
 2100  007661  EM1002:: .ASCIZ /  SKIP SELF-TEST TOOK TOO LONG TO COMPLETE, > 50 MS./
 2101  007666
 2102  007674
 2103  007702
 2104  007710
 2105  007716
 2106  007724
 2107  007732
 2108  007740
 2109  007746  EM1003:: .ASCIZ /  SKIP SELF-TEST COMPLETED TOO SOON, < 10 MS./
 2110  007754
 2111  007762
 2112  007770
 2113  007776
 2114  010004
 2115  010012
 2116  010020
 2117  010024  EM1101:: .ASCIZ /SKIP SELF-TEST TEST/
 2118  010032
 2119  010040
 2120  010046
 2121  010050  EM1201:: .ASCIZ /SELF-TEST TEST/
 2122  010056
 2123  010064
 2124  010067  EM1202:: .ASCIZ /  SELF-TEST TOOK TOO LONG TO COMPLETE, > 3 SECONDS./
 2125  010074
 2126  010102
 2127  010110
 2128  010116
 2129  010124
 2130  010132
 2131  010140
 2132  010146
 2133  010153  EM1203:: .ASCIZ \  SELF-TEST COMPLETED TOO SOON, < 1/2 SECOND.\
 2134  010160
 2135  010166
 2136  010174
 2137  010202
 2138  010210
 2139  010216
 2140  010224
 2141  010231  EM1204:: .ASCIZ /  SELF-TEST DID NOT EXECUTE/
 2142  010236
 2143  010244
```

```
2144  010252
2145  010260
2146  010265  EM1205:: .ASCIZ /  DIAG_FAIL BIT BAD/
2147  010272
2148  010300
2149  010306
2150  010311  EM1301:: .ASCIZ /FAIL SELF-TEST TEST/
2151  010316
2152  010324
2153  010332
2154  010335  EM1302:: .ASCIZ / SELF-TEST ERROR REPORTING BAD/
2155  010342
2156  010350
2157  010356
2158  010364
2159  010372
2160  010374  EM1401:: .ASCIZ /ROM VERSION_NUMBER TEST/
2161  010402
2162  010410
2163  010416
2164  010424  EM1402:: .ASCIZ /  FIFO EMPTY, ONE OR MORE ROM VERSION_NUMBERS MISSING/
2165  010432
2166  010440
2167  010446
2168  010454
2169  010462
2170  010470
2171  010476
2172  010504
2173  010512  EM1403:: .ASCIZ /  ROM VERSION_NUMBER FOUND OUT OF SEQUENCE/
2174  010520
2175  010526
2176  010534
2177  010542
2178  010550
2179  010556
2180  010564
2181  010565  EM1404:: .ASCIZ /  ONE OR MORE ROM VERSION_NUMBERS MISSING/
2182  010572
2183  010600
2184  010606
2185  010614
2186  010622
2187  010630
2188  010636
2189  010637  EM1405:: .ASCIZ /    PROC_1/
2190  010644
2191  010652  EM1406:: .ASCIZ /    PROC_2/
2192  010660
2193  010665  EM1407:: .ASCIZ /NOT FOUND/
2194  010672
2195  010677  EM1408:: .ASCIZ /FOUND/
2196  010704
2197  010705  EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
2198  010712
2199  010720
```

```
2200  010726
2201  010734
2202  010742
2203  010750
2204  010756
2205  010764
2206  010770  EM1604:: .ASCIZ \DEVICE REGISTER WORD READ/WRITE TEST \
2207  010776
2208  011004
2209  011012
2210  011020
2211  011026
2212  011034
2213  011036  EM1701:: .ASCIZ \DEVICE REGISTER WORD READ/MODIFY/WRITE TEST \
2214  011044
2215  011052
2216  011060
2217  011066
2218  011074
2219  011102
2220  011110
2221  011113  EM1801:: .ASCIZ \DEVICE REGISTER BYTE READ/WRITE TEST \
2222  011120
2223  011126
2224  011134
2225  011142
2226  011150
2227  011156
2228  011161  EM1901:: .ASCIZ \DEVICE REGISTER BYTE READ/MODIFY/WRITE TEST \
2229  011166
2230  011174
2231  011202
2232  011210
2233  011216
2234  011224
2235  011232
2236  011236  EM2001:: .ASCIZ /DEVICE STAT REGISTER ID BIT TEST /
2237  011244
2238  011252
2239  011260
2240  011266
2241  011274
2242  011300  EM2002:: .ASCIZ /ID BIT BAD.  EXPECTED: CLEAR, ACTUAL: SET./
2243  011306
2244  011314
2245  011322
2246  011330
2247  011336
2248  011344
2249  011352
2250  011353  EM9009:: .ASCIZ /EXPECTED OR CORRECT/
2251  011360
2252  011366
2253  011374
2254  011377  EM9010:: .ASCIZ /ACTUAL OR MEASURED /
2255  011404
```

```
2256  011412
2257  011420
2258  011423  EM9014:: .ASCIZ /SUMMARY REPORTS FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:/
2259  011430
2260  011436
2261  011444
2262  011452
2263  011460
2264  011466
2265  011474
2266  011502
2267  011510
2268  011516
2269  011517  EM9017:: .ASCII /  FIFO WILL NOT PURGE (DATA.VALID STUCK SET)./
2270  011524
2271  011532
2272  011540
2273  011546
2274  011554
2275  011562
2276  011570
2277  011574          .ASCIZ / REMAINDER OF TEST SKIPPED./
2278  011602
2279  011610
2280  011616
2281  011624
2282  011630  EM9018:: .ASCIZ /NO CODE/
2283  011636
2284  011640  EM9019:: .ASCIZ /NON-SELFTEST/
2285  011646
2286  011654
2287  011655  EM9020:: .ASCIZ /SELFTEST ERROR CODE/
2288  011662
2289  011670
2290  011676
2291  011701  EM9022:: .ASCIZ /DATA CHARACTER/
2292  011706
2293  011714
2294  011720  EM9023:: .ASCIZ /MODEM STATUS CODE/
2295  011726
2296  011734
2297  011742  EM9024:: .ASCIZ /SELFTEST CODE/
2298  011750
2299  011756
2300  011760  EM9026:: .ASCIZ /     LPR CONTENTS: /
2301  011766
2302  011774
2303  012002
2304  012004  EM9301:: .ASCIZ /BMP CODE REPORT/
2305  012012
2306  012020
2307  012024  EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
2308  012032
2309  012040
2310  012046
2311  012054  EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
```

```
 2312  012062
 2313  012070
 2314  012076
 2315  012104
 2316  012112
 2317  012120
 2318  012121  EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
 2319  012126
 2320  012134
 2321  012142
 2322  012150
 2323  012156
 2324  012164
 2325  012172
 2326                  .EVEN
 2327                                .LIST BIN
```

```
2328
2329
2330
2331                            .SBTTL  GLOBAL ERROR REPORT SECTION
2332
2333                            ;++
2334                            ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
2335                            ; USED BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
2336                            ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
2337                            ;--
```

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

012176                          BGNMSG  ER0101
012176                                                                  ER0101::
012176                          SAVE                    ;SAVE THE GPR CONTENTS.
012176  004537  002674                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

012202  032705  000001          BIT     #BIT0,R5        ;TEST FOR READ ERROR.
012206  001410                  BEQ     2$              ;SKIP READ ERROR MSG IF NO READ ERROR.
012210                          PRINTB  #MSG1           ;PRINT READ ERROR MESSAGE.
012210  012746  012302                                          MOV     #MSG1,-(SP)
012214  012746  000001                                          MOV     #1,-(SP)
012220  010600                                                  MOV     SP,R0
012222  104414                                                  TRAP    C$PNTB
012224  062706  000004                                          ADD     #4,SP
012230  032705  000002  2$:     BIT     #BIT1,R5        ;TEST FOR WRITE ERROR.
012234  001410                  BEQ     4$              ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
012236                          PRINTB  #MSG2           ;PRINT WRITE ERROR MESSAGE.
012236  012746  012360                                          MOV     #MSG2,-(SP)
012242  012746  000001                                          MOV     #1,-(SP)
012246  010600                                                  MOV     SP,R0
012250  104414                                                  TRAP    C$PNTB
012252  062706  000004                                          ADD     #4,SP
012256                  4$:     PRINTX  #MSG3           ;SUGGEST THAT DHV MAY BE AT WRONG ADDRESS.
012256  012746  012437                                          MOV     #MSG3,-(SP)
012262  012746  000001                                          MOV     #1,-(SP)
012266  010600                                                  MOV     SP,R0
012270  104415                                                  TRAP    C$PNTX
012272  062706  000004                                          ADD     #4,SP
012276                          PASS                    ;RESTORE THE GPR CONTENTS.
012276  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
012300                          ENDMSG
012300                                                                  L10002:
012300  104423                                                          TRAP    C$MSG
```

```
2394  012302  040445  052502  020123  MSG1::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.%N/
2395  012310  044524  042515  047455
2396  012316  052125  052040  040522
2397  012324  020120  040503  051525
2398  012332  042105  041040  020131
2399  012340  042522  042101  040440
2400  012346  052124  046505  052120
2401  012354  022456  000116
2402  012360  040445  052502  020123  MSG2::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.%N/
2403  012366  044524  042515  047455
2404  012374  052125  052040  040522
2405  012402  020120  040503  051525
2406  012410  042105  041040  020131
2407  012416  051127  052111  020105
2408  012424  052101  042524  050115
2409  012432  027124  047045     000
2410  012437     045  042101  053110  MSG3::  .ASCIZ  /%ADHV MAY BE AT THE WRONG Q-BUS ADDRESS.%N%N/
2411  012444  046440  054501  041040
2412  012452  020105  052101  052040
2413  012460  042510  053440  047522
2414  012466  043516  050440  041055
2415  012474  051525  040440  042104
2416  012502  042522  051523  022456
2417  012510  022516  000116
2418
2419                                  .EVEN
```

```
2420                                    .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0201 -
2421                                    ;*****************************************************************************
2422                                    ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS 2 CONTIGUOUS
2423                                    ;*      ASCII ERROR MESSAGES.  THE ADDRESS OF THE FIRST MESSAGE IS PASSED
2424                                    ;*      AS AN INPUT PARAMETER AND THE ADDRESS OF THE SECOND IS FOUND BY
2425                                    ;*      SEARCHING FOR THE END OF THE FIRST MESSAGE.
2426                                    ;*
2427                                    ;* INPUTS:      R1 - ADDRESS OF THE FIRST MESSAGE TO PRINT.
2428                                    ;*
2429                                    ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
2430                                    ;*
2431                                    ;* CALLING SEQUENCE:    LOAD THE ADDRESS OF THE FIRST MESSAGE IN R1.
2432                                    ;*                      INCLUDE THE LABEL "ER0201" AS THE MESSAGE POINTER
2433                                    ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2434                                    ;*
2435                                    ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2436                                    ;*              THE SECOND MESSAGE SHOULD FOLLOW THE FIRST ONE IN THE PROGRAM
2437                                    ;*              MEMORY.  EACH MESSAGE SHOULD BE DEFINED USING .ASCIZ
2438                                    ;*
2439                                    ;* SUBORDINATE ROUTINES USED: NONE.
2440                                    ;*****************************************************************************
2441
2442  012514                                    BGNMSG  ER0201
2443  012514                                                                              ER0201::
2444  012514                                    SAVE                        ;SAVE THE GPR CONTENTS.
2445  012514  004537  002674                                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2446
2447  012520  010102                             MOV     R1,R2
2448  012522  105722                    2$:      TSTB    (R2)+           ;CHECK FOR A ZERO BYTE (END OF MESSAGE).
2449  012524  001376                             BNE     2$              ;LOOP UNTIL NEXT MESSAGE IS FOUND.
2450
2451  012526                                    PRINTB  #EF0503,R1      ;PRINT THE FIRST MESSAGE.
2452  012526  010146                                                                     MOV     R1,-(SP)
2453  012530  012746  003030                                                             MOV     #EF0503,-(SP)
2454  012534  012746  000002                                                             MOV     #2,-(SP)
2455  012540  010600                                                                     MOV     SP,R0
2456  012542  104414                                                                     TRAP    C$PNTB
2457  012544  062706  000006                                                             ADD     #6,SP
2458  012550                                    PRINTB  #EF0503,R2      ;PRINT THE SECOND MESSAGE.
2459  012550  010246                                                                     MOV     R2,-(SP)
2460  012552  012746  003030                                                             MOV     #EF0503,-(SP)
2461  012556  012746  000002                                                             MOV     #2,-(SP)
2462  012562  010600                                                                     MOV     SP,R0
2463  012564  104414                                                                     TRAP    C$PNTB
2464  012566  062706  000006                                                             ADD     #6,SP
2465
2466  012572                                    PASS                        ;RESTORE THE GPR CONTENTS.
2467  012572  004736                                        JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2468
2469  012574                                    ENDMSG
2470  012574                                                                          L10003:
2471  012574  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.
                        ;*******************************************************************************

012576                          BGNMSG  ER0503
012576                                                                  ER0503::
012576                          PRINTB  #EF0503,R1      ;PRINT THE MESSAGE.
012576  010146                                                              MOV     R1,-(SP)
012600  012746  003030                                                      MOV     #EF0503,-(SP)
012604  012746  000002                                                      MOV     #2,-(SP)
012610  010600                                                              MOV     SP,R0
012612  104414                                                              TRAP    C$PNTB
012614  062706  000006                                                      ADD     #6,SP

012620                          ENDMSG
012620                                                                  L10004:
012620  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.
                        ;*****************************************************************************

012622                          BGNMSG  ER1401
012622                                                          ER1401::

012622                          PRINTB  #EF0503,R1              ;REPORT THE ERROR MESSAGE PASSED IN.
012622  010146                                                          MOV     R1,-(SP)
012624  012746  003030                                                  MOV     #EF0503,-(SP)
012630  012746  000002                                                  MOV     #2,-(SP)
012634  010600                                                          MOV     SP,R0
012636  104414                                                          TRAP    C$PNTB
012640  062706  000006                                                  ADD     #6,SP

                        ;+
                        ; DETERMINE WHICH ROM VERSION NUMBER(S) ARE MISSING.
                        ;-

012644  012705  000143          MOV     #99.,R5                 ;GET INVALID ROM NUMBER.
012650  012701  010637          MOV     #EM1405,R1              ;SELECT PROC_1 MESSAGE.
012654  012702  010665          MOV     #EM1407,R2              ;SELECT THE "NOT FOUND" MESSAGE.
012660  120305                  CMPB    R3,R5                   ;CHECK PROC_1 ROM VERSION NUMBER.
012662  001402                  BEQ     2$                      ;GO REPORT PROC_1 CODE NOT FOUND.
012664  012702  010677          MOV     #EM1408,R2              ;SELECT "FOUND" MESSAGE.
012670  004737  012722  2$:     JSR     PC,50$                  ;GO REPORT MESSAGE.

012674  012701  010652          MOV     #EM1406,R1              ;SELECT PROC_2 MESSAGE.
012700  012702  010665          MOV     #EM1407,R2              ;SELECT THE "NOT FOUND" MESSAGE.
012704  120405                  CMPB    R4,R5                   ;CHECK PROC_2 ROM VERSION NUMBER.
012706  001402                  BEQ     4$                      ;GO REPORT PROC_2 CODE NOT FOUND.
012710  012702  010677          MOV     #EM1408,R2              ;SELECT "FOUND" MESSAGE.
012714  004737  012722  4$:     JSR     PC,50$                  ;GO REPORT THE MESSAGE.
012720  000413                  BR      60$                     ;EXIT.

012722                  50$:    PRINTX  #EF1402,R1,R2           ;REPORT THE MESSAGE.
012722  010246                                                          MOV     R2,-(SP)
```

```
2560  012724  010146                                                    MOV     R1,-(SP)
2561  012726  012746  003137                                            MOV     #EF1402,-(SP)
2562  012732  012746  000003                                            MOV     #3,-(SP)
2563  012736  010600                                                    MOV     SP,R0
2564  012740  104415                                                    TRAP    C$PNTX
2565  012742  062706  000010                                            ADD     #10,SP
2566  012746  000207                    RTS     PC          ;RETURN.
2567  012750                     60$:    ENDMSG
2568  012750                                                L10005:
2569  012750  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
                        ;*******************************************************************************
012752                          BGNMSG  ER1601
012752                                                                  ER1601::
012752  016304  002254          MOV     RMATBB(R3),R4   ;FETCH ADDRESS OF REGISTER NAME MESSAGE.
012756                  PRINTB  #EF1604,R4,R5   ;REPORT BASIC MESSAGE (REG NAME AND LINE #).
012756  010546                                                          MOV     R5,-(SP)
012760  010446                                                          MOV     R4,-(SP)
012762  012746  003324                                                  MOV     #EF1604,-(SP)
012766  012746  000003                                                  MOV     #3,-(SP)
012772  010600                                                          MOV     SP,R0
012774  104414                                                          TRAP    C$PNTB
012776  062706  000010                                                  ADD     #10,SP
013002                  PRINTX  #EF1602,R2      ;PRINT THE EXPECTED DATA.
013002  010246                                                          MOV     R2,-(SP)
013004  012746  003220                                                  MOV     #EF1602,-(SP)
013010  012746  000002                                                  MOV     #2,-(SP)
013014  010600                                                          MOV     SP,R0
013016  104415                                                          TRAP    C$PNTX
013020  062706  000006                                                  ADD     #6,SP
013024                  PRINTX  #EF1603,R1      ;PRINT THE ACTUAL DATA.
013024  010146                                                          MOV     R1,-(SP)
013026  012746  003262                                                  MOV     #EF1603,-(SP)
013032  012746  000002                                                  MOV     #2,-(SP)
013036  010600                                                          MOV     SP,R0
013040  104415                                                          TRAP    C$PNTX
013042  062706  000006                                                  ADD     #6,SP
013046                  ENDMSG
013046                                                          L10006:
013046  104423                                                          TRAP    C$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.
                        ;*****************************************************************************
013050                          BGNMSG  ER1603
013050                                                                  ER1603::
013050                          SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
013050  004537  002674                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
013054
013054                           PRINTB #EF0503,R1      ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
013054  010146                                                          MOV     R1,-(SP)
013056  012746  003030                                                  MOV     #EF0503,-(SP)
013062  012746  000002                                                  MOV     #2,-(SP)
013066  010600                                                          MOV     SP,R0
013070  104414                                                          TRAP    C$PNTB
013072  062706  000006                                                  ADD     #6,SP
013076  013702  002670              MOV     ERRMSG,R2       ;GET THE "TEST MESSAGE".
013102                           PRINTB #EF1601,R2      ;PRINT "TEST ABORTED" MESSAGE.
013102  010246                                                          MOV     R2,-(SP)
013104  012746  003174                                                  MOV     #EF1601,-(SP)
013110  012746  000002                                                  MOV     #2,-(SP)
013114  010600                                                          MOV     SP,R0
013116  104414                                                          TRAP    C$PNTB
013120  062706  000006                                                  ADD     #6,SP
013124                          PASS                    ;RESTORE THE CONTENTS OF THE GPRS.
013124  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
013126                          ENDMSG
013126                                                                  L10007:
013126  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.
                        ;*******************************************************************************

                            BGNMSG  ER9004
                                                                        ER9004::

                            PRINTB  #EF0503,#EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
013130  012746  011423                                                      MOV     #EM9014,-(SP)
013134  012746  003030                                                      MOV     #EF0503,-(SP)
013140  012746  000002                                                      MOV     #2,-(SP)
013144  010600                                                              MOV     SP,R0
013146  104414                                                              TRAP    C$PNTB
013150  062706  000006                                                      ADD     #6,SP
013154  005002                    CLR     R2              ;CLEAR THE LINE COUNTER.
013156  013703  002420            MOV     ERSMRF,R3       ;GET THE ERROR SUMMARY FLAGS.
013162  005004                    CLR     R4              ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
013164  000241            2$:     CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
013166  006003                    ROR     R3              ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
013170  103013                BCC     4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
013172                          PRINTX  #EF9010,R2,ERCNTB(R4)
013172  016446  002422                                                      MOV     ERCNTB(R4),-(SP)
013176  010246                                                              MOV     R2,-(SP)
013200  012746  003730                                                      MOV     #EF9010,-(SP)
013204  012746  000003                                                      MOV     #3,-(SP)
013210  010600                                                              MOV     SP,R0
013212  104415                                                              TRAP    C$PNTX
013214  062706  000010                                                      ADD     #10,SP
013220  012405            4$:     MOV     (R4)+,R5        ;INCREMENT THE LINE OFFSET BY 2.
013222  005202                    INC     R2              ;INCREMT THE LINE COUNTER.
013224  005703                    TST     R3              ;CHECK THE ERROR SUMMARY FLAGS.
013226  001356                BNE     2$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.

                            ENDMSG
                                                                    L10010:
013230  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.
                        ;*******************************************************************************

                            BGNMSG  ER9007
013232                                                          ER9007::
013232

013232  042703  177760      BIC     #177760,R3      ;REMOVE ALL BUT LINE # BITS FROM LINE # WORD.
013236                      PRINTB  #EF9018,R1,R3   ;REPORT SECONDARY ERROR MESSAGE.
013236  010346                                                      MOV     R3,-(SP)
013240  010146                                                      MOV     R1,-(SP)
013242  012746  004200                                              MOV     #EF9018,-(SP)
013246  012746  000003                                              MOV     #3,-(SP)
013252  010600                                                      MOV     SP,R0
013254  104414                                                      TRAP    C$PNTB
013256  062706  000010                                              ADD     #10,SP
013262                      PRINTX  #EF9017,R1,R2   ;REPORT THE ACTUAL INCORRECT CODE.
013262  010246                                                      MOV     R2,-(SP)
013264  010146                                                      MOV     R1,-(SP)
013266  012746  004124                                              MOV     #EF9017,-(SP)
013272  012746  000003                                              MOV     #3,-(SP)
013276  010600                                                      MOV     SP,R0
013300  104415                                                      TRAP    C$PNTX
013302  062706  000010                                              ADD     #10,SP

013306                      ENDMSG
013306                                                  L10011:
013306  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.
                        ;*******************************************************************************

013310                          BGNMSG  ER9008
013310                                                          ER9008::

                        ;+
                        ; EXTRACT THE LINE NUMBER FROM THE INCORRECT CODE OR CHARACTER WHICH WAS READ
                        ;    FROM THE SELFTEST CODE FIFO SLOT.
                        ;-

013310  010203                  MOV     R2,R3
013312  000303                  SWAB    R3
013314  042703  177760          BIC     #177760,R3      ;CALCULATE LINE NUMBER OF CODE.
013320                  PRINTB  #EF9016,R1,R3   ;REPORT TYPE OF INCORRECT CODE FOUND.
013320  010346                                                  MOV     R3,-(SP)
013322  010146                                                  MOV     R1,-(SP)
013324  012746  004027                                         MOV     #EF9016,-(SP)
013330  012746  000003                                         MOV     #3,-(SP)
013334  010600                                                  MOV     SP,R0
013336  104414                                                  TRAP    C$PNTB
013340  062706  000010                                         ADD     #10,SP
013344                  PRINTX  #EF9017,R1,R2   ;REPORT THE ACTUAL INCORRECT CODE.
013344  010246                                                  MOV     R2,-(SP)
013346  010146                                                  MOV     R1,-(SP)
013350  012746  004124                                         MOV     #EF9017,-(SP)
013354  012746  000003                                         MOV     #3,-(SP)
013360  010600                                                  MOV     SP,R0
013362  104415                                                  TRAP    C$PNTX
013364  062706  000010                                         ADD     #10,SP

013370                  ENDMSG
013370                                                  L10012:
013370  104423                                                  TRAP    C$MSG
```

```
2818                                    .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9301 -
2819                                    ;*************************************************************************
2820                                    ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ANY BMP CODES
2821                                    ;*      THAT ARE FOUND IN THE BMP CODE QUEUE, TOGETHER WITH THE THE NUMBER OF
2822                                    ;*      THE TEST THAT WAS EXECUTING AT THE TIME THE BMP CODE WAS LOGGED.
2823                                    ;*
2824                                    ;* INPUTS:      R1 - THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
2825                                    ;*              R2 - THE ADDRESS OF THE NEXT EMPTY CELL IN THE QUEUE.
2826                                    ;*
2827                                    ;* OUTPUTS:     THE TEST NUMBER FOLLOWED BY THE BMP CODE ARE PRINTED AT THE
2828                                    ;*              OPERATOR CONSOLE.
2829                                    ;*
2830                                    ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9301" AS THE MESSAGE POINTER
2831                                    ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2832                                    ;*
2833                                    ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2834                                    ;*
2835                                    ;* SUBORDINATE ROUTINES USED: NONE.
2836                                    ;*************************************************************************
2837
2838  013372                                BGNMSG  ER9301
2839  013372                                                                          ER9301::
2840  013372                                SAVE                    ;SAVE THE GPRS ON THE STACK.
2841  013372  004537  002674                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2842
2843  013376                                PRINTB  #EF0503,R1      ;REPORT UNEXPECTED BMP CODES FOUND.
2844  013376  010146                                                                 MOV     R1,-(SP)
2845  013400  012746  003030                                                        MOV     #EF0503,-(SP)
2846  013404  012746  000002                                                        MOV     #2,-(SP)
2847  013410  010600                                                                MOV     SP,R0
2848  013412  104414                                                                TRAP    C$PNTB
2849  013414  062706  000006                                                        ADD     #6,SP
2850  013420  012703  002464                        MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
2851  013424  012705  012024                        MOV     #EM9302,R5      ;GET THE MESSAGE TO BE REPORTED.
2852  013430  012301                        2$:     MOV     (R3)+,R1        ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
2853  013432  012304                        MOV     (R3)+,R4        ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
2854  013434  004737  013516                JSR     PC,50$          ;GO REPORT THE BMP CODE.
2855  013440  020302                        CMP     R3,R2           ;CHECK IF ALL CODES HAVE BEEN REPORTED.
2856  013442  103772                        BLO     2$              ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
2857                                    ;+
2858                                    ; CHECK IF OVERFLOW HAS OCCURRED.
2859                                    ;  THE CONDITIONS FOR OVERFLOW ARE: THE POINTER CONTAINS THE ADDRESS OF THE
2860                                    ;  LAST CELL IN THE QUEUE, AND A BMP CODE HAS ALREADY BEEN WRITTEN INTO THAT
2861                                    ;  CELL.
2862                                    ;-
2863  013444  020227  002660                        CMP     R2,#BMPCQE-4    ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
2864  013450  001036                        BNE     60$             ;EXIT IF NOT AT THE LAST LOCATION.
2865  013452  005762  000002                TST     2(R2)           ;CHECK FOR A BMP CODE IN THE LAST CELL
2866  013456  001433                        BEQ     60$             ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
2867  013460  012301                        MOV     (R3)+,R1        ;GET THE TEST NUMBER OFF THE QUEUE.
2868  013462  011304                        MOV     (R3),R4         ;GET THE BMP CODE OFF THE QUEUE.
2869  013464  012705  012054                MOV     #EM9303,R5      ;SELECT THE MESSAGE TO BE REPORTED.
2870  013470                                PRINTX  #EF9302         ;REPORT OVERFLOW CONDITION.
2871  013470  012746  004407                                                        MOV     #EF9302,-(SP)
2872  013474  012746  000001                                                        MOV     #1,-(SP)
2873  013500  010600                                                                MOV     SP,R0
```

```
2874  013502  104415                                                          TRAP    C$PNTX
2875  013504  062706  000004                                                  ADD     #4,SP
2876  013510  004737  013516          JSR     PC,50$          ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
2877  013514  000414                  BR      60$             ;EXIT.
2878
2879  013516                  50$:    PRINTX  #EF9301,R5,R1,R4  ;PRINT THE MESSAGE.
2880  013516  010446                                                          MOV     R4,-(SP)
2881  013520  010146                                                          MOV     R1,-(SP)
2882  013522  010546                                                          MOV     R5,-(SP)
2883  013524  012746  004331                                                  MOV     #EF9301,-(SP)
2884  013530  012746  000004                                                  MOV     #4,-(SP)
2885  013534  010600                                                          MOV     SP,R0
2886  013536  104415                                                          TRAP    C$PNTX
2887  013540  062706  000012                                                  ADD     #12,SP
2888  013544  000207                  RTS     PC              ;RETURN.
2889  013546                  60$:    PASS                    ;RESTORE THE GPR CONTENTS.
2890  013546  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2891
2892  013550                  ENDMSG
2893  013550                                                          L10013:
2894  013550  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL SUBROUTINES SECTION


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

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

013552                  CALMSL:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
013552  004537  002674          JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
013556  005037  013772          CLR     62$             ;CLEAR THE 2ND TIME FLAG.
                        ;+
                        ; SYNCHRONIZE WITH THE LTC.
                        ;-
013562  012705  000001  2$:     MOV     #1,R5           ;SET OUTER LOOP COUNTER TO 1 LOOP.
                                                        ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE <**
                                                        ;FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <**
013566  005000                  CLR     R0              ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
013570  012737  000001  002322  MOV     #1,TIMER1       ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
013576  005737  002322  4$:     TST     TIMER1          ;CHECK FOR COUNTER HAVING GONE TO ZERO.
013602  001410                  BEQ     6$              ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
013604  005200                  INC     R0              ;COUNT THIS ITERATION OF THE INNER LOOP.
013606  001373                  BNE     4$              ;LOOP IF COUNTER HAS NOT TURNED OVER.
013610  005305                  DEC     R5              ;DECREMENT THE INNER LOOP COUNTER.
013612  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.
                        ;-
013614  005037  002320          CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
013620  000241                  CLC                     ;INDICATE FAILURE FOR RETURN.
013622  000461                  BR      60$             ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
                        ;+
                        ; WE ARE NOW SYNCHRONIZED WITH THE LTC.
                        ; SET UP FOR THE CALIBRATION LOOP.
                        ;-
```

```
 2958  013624  012704  002322          6$:     MOV     #TIMER1,R4      ;WILL TEST TIMER1 IN THE LOOP BELOW.
 2959  013630  005001                          CLR     R1              ;CLEAR THE OUTER LOOP COUNTER.
 2960  013632  005002                          CLR     R2              ;INDICATE TO CHECK ALL BITS OF TIMER1.
 2961  013634  005003                          CLR     R3              ;INDICATE TO CHECK FOR TIMER1 CLEAR.
 2962  013636  012714  000001                  MOV     #1,(R4)         ;LOAD TIMER1 WITH COUNT OF 1.
 2963
 2964  013642  013705  002334          8$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
 2965  013646  011400                  10$:    MOV     (R4),R0         ;GET THE TIMER1 VALUE.
 2966  013650  010037  013774                  MOV     R0,64$          ;SAVE WORD (LIKE IN THE REAL LOOP).
 2967  013654  040200                          BIC     R2,R0           ;LEAVE ALL THE BITS.
 2968  013656  020003                          CMP     R0,R3           ;COMPARE AGAINST ZERO.
 2969  013660  000261                          SEC                     ;SET CARRY IN CASE OF SUCCESS.
 2970  013662  001406                          BEQ     12$             ;EXIT LOOP IF TIMER1 HAS CLEARED.
 2971  013664  005305                          DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
 2972  013666  001367                          BNE     10$             ;LOOP IF MS NOT UP.
 2973  013670  005301                          DEC     R1              ;DECREMENT THE MS TIME COUNT.
 2974  013672  001363                          BNE     8$              ;KEEP LOOPING.
 2975  013674  004737  014316                  JSR     PC,OOPS         ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
 2976                                  ;+
 2977                                  ; WE HAVE NOW HAVE LOOP COUNT INFORMATION FOR ONE CLOCK TICK.
 2978                                  ; WE HAVE NEGATIVE OF NUMBER OF OUTER LOOPS IN R1, EACH IS MSLCNT INNER LOOPS.
 2979                                  ; WE HAVE THE PORTION OF THE LAST OUTER LOOP NOT EXECUTED, IN R5.
 2980                                  ; NOW WE CALCULATE THE TOTAL NUMBER OF INNER LOOPS EXECUTED.
 2981                                  ;-
 2982  013700  005401                  12$:    NEG     R1              ;GET NUMBER OF OUTER LOOPS.
 2983  013702  013702  002334                  MOV     MSLCNT,R2       ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
 2984  013706  010203                          MOV     R2,R3           ;COPY NUMBER OF LOOPS FOR MULTIPLY.
 2985  013710  160502                          SUB     R5,R2           ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
 2986  013712  010204                          MOV     R2,R4           ;  AND ADD TO ACCUMULATOR LSWORD.
 2987  013714  005005                          CLR     R5              ;CLEAR ACCUMULATOR MSWORD.
 2988  013716  005301                  14$:    DEC     R1              ;CHECK R1 FOR 0 CONDITION
 2989  013720  100403                          BMI     16$             ; SKIP MULTIPLICATION IF ZERO
 2990  013722  060304                          ADD     R3,R4           ;MULTIPLY NUMBER OF INNER
 2991  013724  005505                          ADC     R5              ; LOOPS PER OUTER LOOP BY
 2992  013726  000773                          BR      14$             ;NUMBER OF OUTER LOOPS PERFORMED.
 2993                                  ;+
 2994                                  ; DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER LTC TICK.
 2995                                  ;-
 2996  013730  013701  002332          16$:    MOV     MSTICK,R1       ;# OF MS PER LTC TICK IS DIVISOR.
 2997  013734  010403                          MOV     R4,R3           ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
 2998  013736  010502                          MOV     R5,R2           ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
 2999  013740  004737  016340                  JSR     PC,UNSDIV       ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
 3000  013744  103402                          BCS     18$             ;BYPASS OOPS IF WE'RE OK.
 3001  013746  004737  014316                  JSR     PC,OOPS         ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
 3002  013752  010137  002334          18$:    MOV     R1,MSLCNT       ;SET NEW VALUE FOR MS LOOP COUNT.
 3003  013756  005137  013772                  COM     62$             ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
 3004  013762  001277                          BNE     2$              ;BRANCH IF ONLY ONE ITERATION DONE.
 3005  013764  000261                          SEC                     ;SET THE SUCCESS FLAG FOR EXIT.
 3006
 3007  013766                          60$:    PASS                    ;RESTORE GPRS,
 3008  013766  004736                                  JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
 3009  013770  000207                          RTS     PC              ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
 3010
 3011  013772  000000          62$:    .WORD   0               ;2ND CALIBRATION ITERATION FLAGS.
 3012  013774  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/O LOCATION.
                                ;*      IF THE TRAP DOES NOT OCCUR, THIS ROUTINE RETURNS A SUCCESS INDICATION.
                                ;*
                                ;* INPUTS:      R0 - SOURCE ADDRESS FOR MOVE.
                                ;*              R1 - DESTINATION ADDRESS FOR MOVE.
                                ;*              (R0) - SOURCE FOR THE MOVE.
                                ;*
                                ;* OUTPUTS:     (R1) - WRITTEN TO THE CONTENTS OF (R0).
                                ;*              CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED.
                                ;*              TP4FLG - NONZERO IF TRAP OCCURRED, CLEARED OTHERWISE.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,CKTRAP
                                ;*
                                ;* COMMENTS:    IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS WHICH
                                ;*              IS LABELED ADRPTR WILL BE THE TRAP PC ADDRESS ON THE STACK.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;*************************************************************************

013776                          CKTRAP:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
013776  004537  002674                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014002  005037  002304                CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAGS.
014006  011011                        MOV     (R0),(R1)       ;PERFORM THE MOVE IN QUESTION.
014010  005737  002304        ADRPTR:: TST    TP4FLG          ;CHECK FOR OCCURENCE OF TRAP.
014014  000261                        SEC                     ;INDICATE SUCCESS.
014016  001401                        BEQ     60$             ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
014020  000241                        CLC                     ;INDICATE FAILURE.
014022                          60$:    PASS                    ;RESTORE GPRS.
014022  004736                                        JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014024  000207                        RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - CLR16W -
                        ;++ ************************************************************************
                        ;*                        - CLEAR SIXTEEN WORDS ROUTINE -
                        ;*      THIS SUBROUTINE CLEARS 16 WORDS STARTING WITH THE SPECIFIED WORD.
                        ;*
                        ;* INPUTS:      R0 - ADDRESS OF THE FIRST WORD TO CLEAR.
                        ;*
                        ;* OUTPUTS:     (R0) TO (R0+15) - 16 WORDS OF MEMORY ARE CLEARED TO 0.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CLR16W
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

014026                  CLR16W:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014026  004537  002674                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014032  012701  000020          MOV     #16.,R1         ;SET THE LOOP COUNTER TO 16.
014036  005020          2$:     CLR     (R0)+           ;CLEAR A WORD OF MEMORY.
014040  005301          DEC     R1              ;COUNT THIS LOOP.
014042  001375          BNE     2$              ;LOOP IF NOT 16 WORD CLEARED.
014044                  60$:    PASS                    ;RESTORE GPRS.
014044  004736                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014046  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.
                        ;-- ************************************************************************

014050                  CNTERR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
014050  004537  002674                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; COUNT THE ERROR ON THE COUNTER FOR THE SPECIFIED LINE.
                        ;-
014054  006305                   ASL     R5                     ;FORM WORD OFFSET FROM LINE NUMBER.
014056  016501  002422           MOV     ERCNTB(R5),R1          ;GET THE PRESENT ERROR COUNT FOR THIS LINE.
014062  005201                   INC     R1                     ;COUNT ERROR.
014064  103402                  BCS     2$                      ;OVERFLOW?  YES, DON'T UPDATE COUNTER IN TABLE.
014066  010165  002422           MOV     R1,ERCNTB(R5)          ;UPDATE ERROR COUNTER TABLE ENTRY.
014072  005737  002206  2$:      TST     NDERPT
014076  001411                  BEQ     60$                     ;SUMMARYS DISABLED?  YES, EXIT WITH CARRY 0.
014100  020137  002206           CMP     R1,NDERPT              ;NO,  CHECK FOR ENOUGH ERRORS FOR SUMMARY USE.
014104  101002                  BHI     4$                      ;ENOUGH ERRORS TO USE SUMMARY?  YES, GO HANDLE.
014106  000241                   CLC                            ;INDICATE NOT TO USE SUMMARY REPORT YET.
014110  000404                  BR      60$                     ;EXIT WITH CARRY 0.
014112  056537  002346  002420  4$:      BIS     BITTBL(R5),ERSMRF       ;SET THE ERROR SUMMARY FLAG FOR LINE.
014120  000261                   SEC                            ;INDICATE TO USE SUMMARY REPORT.
014122                          60$:    PASS                    ;RESTORE GPRS.
014122  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014124  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.
                                ;*****************************************************************************

014126                          DELAY:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
014126  004537  002674                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
014132  010401                          MOV     R4,R1           ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
014134  012702  177777                  MOV     #-1,R2          ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
014140  005003                          CLR     R3              ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
014142  012704  014164                  MOV     #62$,R4         ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
014146  004737  014302                  JSR     PC,MSLOOP       ;DELAY THE REQUESTED # OF MS.
014152  103002                          BCC     60$             ;EXIT ROUTINE IF WE TIMED-OUT.]
014154  004737  014316                  JSR     PC,OOPS         ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
014160                          60$:    PASS                    ;RESTORE GPRS.
014160  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014162  000207                          RTS     PC

014164  177777                  62$:    .WORD   -1              ;DUMMY, NON-ZERO WORD.
```

```
3148                            .SBTTL  GLOBAL SUBROUTINE                - MSLGET -
3149                            ;*************************************************************************
3150                            ;*      - MILLI SECONDS LOOP WHICH RETURNS READ WORD AND REMAINING TIME -
3151                            ;*      THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE.  IT IS USED
3152                            ;*      TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD.  THE
3153                            ;*      CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
3154                            ;*      DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
3155                            ;*      THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
3156                            ;*      ROUTINE AND THEN ONCE EACH MILLI-SECOND THERE AFTER.
3157                            ;*      UPON RETURN, THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION
3158                            ;*      IS RETURNED BY THIS SUBROUTINE.
3159                            ;*
3160                            ;* INPUTS:      R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
3161                            ;*              R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
3162                            ;*              R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
3163                            ;*              R4 - ADDRESS OF THE WORD TO TEST.
3164                            ;*              MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
3165                            ;*
3166                            ;* OUTPUTS:     R0 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
3167                            ;*              R1 - REMAINING NUMBER OF MS IN TIME-OUT TIME.
3168                            ;*              CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
3169                            ;*
3170                            ;* CALLING SEQUENCE:    JSR     PC,MSLGET
3171                            ;*
3172                            ;* COMMENTS:    THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
3173                            ;*              CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
3174                            ;*              ON THE SYSTEM.
3175                            ;*              THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
3176                            ;*              DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
3177                            ;*              LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
3178                            ;*              IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
3179                            ;*              THE DESIRED CONDITION BEFORE RETURNING.  IT INDICATES SUCCESS
3180                            ;*              IF THE CONDITION IS MET, FAILURE OTHERWISE.
3181                            ;*
3182                            ;*
3183                            ;* SUBORDINATE ROUTINES CALLED: NONE.
3184                            ;*************************************************************************
3185
3186  014166                    MSLGET:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
3187  014166  004537  002674                   JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3188                            ;+
3189                            ; SET UP MASK FOR REMOVING UNUSED BITS IN THE TEST WORD, AND CLEAR UNUSED
3190                            ; BITS IN THE DESIRED STATE WORD TO ALLOW DIRECT COMPARISON.
3191                            ;-
3192  014172  005102                   COM     R2              ;GET MASK OF UNUSED BITS.
3193  014174  040203                   BIC     R2,R3           ;MASK OUT UNUSED BITS IN DESIRED STATE WORD.
3194                            ;+
3195                            ; HANDLE THE TEST AND EXIT IF WE HAVE A 0 TIME-OUT VALUE.
3196                            ;-
3197  014176  005701                   TST     R1              ;TEST THE TIME-OUT VALUE FOR ZERO.
3198  014200  001011                   BNE     2$              ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
3199  014202  011400                   MOV     (R4),R0         ;GET THE WORD TO TEST BEFORE EXITING.
3200  014204  010037  014300           MOV     R0,62$          ;SAVE VALUE SO WE CAN RETURN IT.
3201  014210  040200                   BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
3202  014212  020003                   CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
3203  014214  000261                   SEC                     ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
```

```
3204  014216  001420                        BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
3205  014220  000241                        CLC                     ;INDICATE FAILURE (TIME-OUT).
3206  014222  000416                        BR      6$              ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
3207                                ;+
3208                                ; NON-ZERO TIME-OUT VALUE.  LOOP, WAITING FOR CONDITION OR TIME-OUT.
3209                                ;-
3210  014224  013705  002334       2$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
3211  014230  011400                4$:     MOV     (R4),R0         ;GET THE WORD TO TEST.
3212  014232  010037  014300               MOV     R0,62$          ;SAVE WORD IN CASE THIS IS THE LAST.
3213  014236  040200                        BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
3214  014240  020003                        CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
3215  014242  000261                        SEC                     ;SET CARRY IN CASE OF SUCCESS.
3216  014244  001405                        BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
3217  014246  005305                        DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
3218  014250  001367                        BNE     4$              ;LOOP IF MS NOT UP.
3219  014252  005301                        DEC     R1              ;DECREMENT THE MS TIME COUNT.
3220  014254  001363                        BNE     2$              ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
3221  014256  000241                        CLC                     ;CLEAR CARRY, WE TIMED-OUT.
3222                                ;+
3223                                ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME-OUT VALUE).
3224                                ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD.  EXIT ROUTINE.
3225                                ;-
3226  014260  013700  014300       6$:     MOV     62$,R0          ;PASS OUT THE LAST READ WORD.
3227  014264                        60$:    PASS    R0,R1           ;RESTORE GPRS, EXCEPT THE FOLLOWING:
3228  014264  010066  000002                       MOV     R0,R0SLOT(SP)           ;PUT R0 IN STACK SLOT.
3229  014270  010166  000004                       MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
3230  014274  004736                               JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3231                                                 ;R0 - LAST READ WORD CHECKED FOR CONDITION.
3232                                                 ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
3233  014276  000207                        RTS     PC              ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
3234                                ;+
3235                                ; LOCAL STORAGE.
3236                                ;-
3237  014300  000000                62$:    .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.
                        ;*****************************************************************************

014302                  MSLOOP:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
014302  004537  002674                  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.
                        ;-
014306  004737  014166          JSR     PC,MSLGET       ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.

014312                  60$:    PASS                    ;RESTORE GPRS,
014312  004736                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014314  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.
                        ;-- ************************************************************************

014316                  OOPS::  SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
014316  004537  002674                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                            ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
014322                          ERRSF   101,EM0101
014322  104454                                                                  TRAP    C$ERSF
014324  000145                                                                  .WORD   101
014326  014362                                                                  .WORD   EM0101
014330  000000                                                                  .WORD   0
                            ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
014332                          PRINTF  #EM0102
014332  012746  014446                                                          MOV     #EM0102,-(SP)
014336  012746  000001                                                          MOV     #1,-(SP)
014342  010600                                                                  MOV     SP,R0
014344  104417                                                                  TRAP    C$PNTF
014346  062706  000004                                                          ADD     #4,SP
014352                  2$:     BREAK                   ;LOOK FOR OPERATOR CONTROL-C INPUT.
014352  104422                                                                  TRAP    C$BRK
014354  000776                  BR      2$              ;INFINITE LOOP.
014356                  60$:    PASS                    ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
014356  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
014360  000207                  RTS     PC              ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.

014362  047510  052123  041440  EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
014370  046517  052520  042524
014376  020122  040510  042122
014404  040527  042522  047440
014412  020122  047523  052106
014420  040527  042522  041040
014426  043525  042440  041516
014434  052517  052116  051105
014442  042105  000056
014446  047045  040445  051120  EM0102:: .ASCIZ /%N%APROGRAM HUNG, WAITING FOR A CONTROL-C. <**************%N%N/
014454  043517  040522  020115
014462  052510  043516  020054
014470  040527  052111  047111
014476  020107  047506  020122
014504  020101  047503  052116
```

```
3339  014512  047522  026514  027103
3340  014520  036040  025052  025052
3341  014526  025052  025052  025052
3342  014534  025052  022452  022516
3343  014542  000116
3344                                  .EVEN
```

```
3345                            .SBTTL  GLOBAL SUBROUTINE                - PUFIFO -
3346                            ;*****************************************************************************
3347                            ;*      - PURGE THE FIFO
3348                            ;*      THIS ROUTINE TRIES TO REMOVE ALL THE CHARACTERS FROM THE FIFO.
3349                            ;*      ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE BMP CODE QUEUE.
3350                            ;*
3351                            ;* INPUTS:      RBUFA- CONTAINS THE  ADDRESS OF THE RECEIVER.
3352                            ;*
3353                            ;*
3354                            ;* OUTPUTS:     CARRY BIT - INDICATES THE STATE OF THE FIFO, SET:= PURGED.
3355                            ;*              BMPCQ - THE CONTENTS OF THE BMP CODE QUEUE MAY BE UPDATED.
3356                            ;*
3357                            ;* CALLING SEQUENCE:    JSR     PC,PUFIFO
3358                            ;*
3359                            ;* COMMENTS:
3360                            ;*
3361                            ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
3362                            ;*****************************************************************************
3363
3364  014544                    PUFIFO::SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
3365  014544  004537  002674                    JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3366  014550  012701  001000            MOV     #512.,R1                ;SET MAXIMUM TRY COUNT OF 512.
3367  014554  013704  002216            MOV     RBUFA,R4                ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
3368
3369  014560  011402                    2$:     MOV     (R4),R2         ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
3370  014562  100016                    BPL     6$                      ;EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.
3371                                    ;+
3372                                    ; CHECK IF THE READ CHARACTER IS ACTUALLY A BMP CODE.
3373                                    ;  IF IT IS, THEN SAVE IT ON THE BMP CODE QUEUE TO BE REPORTED LATER.
3374                                    ;-
3375  014564  012700  070000            MOV     #70000,R0               ;GENERATE A BIT MAP OF CHAR ERROR BITS
3376  014570  040200                    BIC     R2,R0                   ; WHICH ARE NOT SET FOR CHAR.
3377  014572  001006                    BNE     4$                      ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
3378                                  ;+
3379                                  ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
3380                                  ;-
3381  014574  012700  000300            MOV     #300,R0                 ; CHECK IF BMP OR SELFTEST?.
3382  014600  040200                    BIC     R2,R0                   ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
3383  014602  001002                    BNE     4$                      ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
3384  014604  004737  016032            JSR     PC,SAVBMP               ;SAVE BMP CODE ON THE QUEUE.
3385
3386  014610  005301                    4$:     DEC     R1              ;DECREMENT THE TRY COUNT.
3387  014612  001362                    BNE     2$                      ;LOOP TO TRY AGAIN.
3388  014614  000241                     CLC                            ;CLEAR CARRY,TO INDICATE FIFO NOT PURGED.
3389  014616  000401                    BR      60$                     ;EXIT WITH CARRY CLEAR.
3390  014620  000261                    6$:     SEC                     ;SET CARRY, TO INDICATE FIFO PURGED.
3391
3392  014622                            60$:    PASS                    ;RESTORE GPRS,
3393  014622  004736                                    JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
3394                                                                    ;CARRY BIT, SET INDICATES FIFO PURGED.
3395  014624  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.
                        ;*              LPRO - 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.
                        ;*              TXBFCO - 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.
                        ;-- *****************************************************************************

014626                  RDPDR:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
014626  004537  002674                  JSR     R5,$REGS0       ;CALL REGISTER SAVE SUBRT.
014632  012737  012752  002672          MOV     #ER1601,ERRBLK  ;SET UP THE ADDRESS OF THE ERROR REPORT RTN.
                        ;+
                        ; DETERMINE WHETHER REGISTER DATA SHOULD BE INVERTED FROM DATA PATTERN.
                        ;-
014640  005704                  TST     R4                      ;CHECK THE OPERAND TYPE INDICATOR.
014642  100001                BPL     2$                      ;BIC WRITE PERFORMED?  NO, USE STANDARD DATA.
014644  005102                  COM     R2                      ;YES, INVERT THE DATA PATTERN.
                        ;+
                        ; SET UP OUTER LOOP.
                        ;-
014646  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.
```

```
3452                                    ;-
3453  014650  010237  015044            4$:     MOV     R2,70$          ;SAVE THE OUTER LOOP DATA PATTERN.
3454  014654  010577  165334                    MOV     R5,@CSRA        ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
3455  014660  010500                            MOV     R5,R0
3456  014662  006300                            ASL     R0
3457  014664  036037  002346  002210            BIT     BITTBL(R0),ACTLNS
3458  014672  001452                          BEQ     16$             ;IS THE LINE ACTIVE?  NO, SKIP THE LINE.
3459  014674  012703  000004                    MOV     #LPR0,R3        ;YES, INITIALIZE REGISTER OFFSET FOR LPR.
3460                                    ;+
3461                                    ;       THE INNER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP READS AND COMPARES
3462                                    ;       DATA FROM A DEVICE REGISTER.
3463                                    ;-
3464  014700  010204                    6$:     MOV     R2,R4           ;SAVE THE INNER LOOP DATA PATTERN.
3465  014702  046302  002234                    BIC     UNBTTB(R3),R2   ;REMOVE UNUSED BITS FROM EXPECTED DATA.
3466  014706  016300  002214                    MOV     DRADRT(R3),R0
3467  014712  005766  000010                    TST     R3SLOT(SP)      ;CHECK THE ACCESS TYPE INDICATOR.
3468  014716  001002                          BNE     8$              ;BYTE ACCESS?  YES, GO PERFORM BYTE READ.
3469  014720  011001                            MOV     (R0),R1         ;NO, PERFORM WORD READ OF DEVICE REGISTER.
3470  014722  000416                          BR      12$
3471  014724  100410                    8$:     BMI     10$             ;LOW BYTE ACCESS?  YES, GO DO LOW BYTE READ.
3472  014726  005200                            INC     R0              ;HIGH BYTE ACCESS.  FORM HIGH BYTE ADDRESS.
3473  014730  111001                            MOVB    (R0),R1         ;READ THE HI BYTE OF THE DUT REGISTER.
3474  014732  000301                            SWAB    R1              ;PUT HI BYTE BACK INTO THE HI BYTE.
3475  014734  042701  000377                    BIC     #377,R1         ;REMOVE THE UNUSED BYTE IN ACTUAL DATA.
3476  014740  042702  000377                    BIC     #377,R2         ;REMOVE THE UNUSED BYTE IN EXPECTED DATA.
3477  014744  000405                          BR      12$
3478  014746  111001                    10$:    MOVB    (R0),R1         ;READ THE LOW BYTE OF THE DUT REGISTER.
3479  014750  042701  177400                    BIC     #177400,R1      ;REMOVE THE UNUSED BYTE.
3480  014754  042702  177400                    BIC     #177400,R2      ;FORM EXPECTED LOW BYTE FOR COMAPARISON.
3481
3482  014760  046301  002234            12$:    BIC     UNBTTB(R3),R1   ;REMOVE UNUSED BITS FROM ACTUAL DATA.
3483  014764  020102                            CMP     R1,R2           ;COMPARE ACTUAL AND EXPECTED DATA.
3484  014766  001404                          BEQ     14$             ;ACTUAL = EXPECTED?  YES, SKIP ERROR.
3485  014770  004737  014050                  JSR     PC,CNTERR       ;NO, COUNT THE ERROR, CHECK FOR ERROR SUMMARY.
3486  014774  103401                          BCS     14$             ;USE ERROR SUMMARY?  YES, SKIP ERROR.
3487                                      ;NO, REPORT 'BAD BIT(S) IN DEVICE XXXXX REGISTER FOR LINE NN (D)."
3488  014776                                  ERROR
3489  014776  104460                                                                  TRAP    C$ERROR
3490  015000  010402                    14$:    MOV     R4,R2           ;RESTORE THE INNER LOOP DATA PATTERN.
3491  015002  004737  015416                  JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
3492  015006  062703  000002                    ADD     #2,R3           ;SET REGISTER OFFSET TO THE NEXT REGISTER.
3493  015012  020327  000016                    CMP     R3,#TXBFCO      ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
3494  015016  003730                          BLE     6$              ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
3495                                    ;+
3496                                    ; BACK INTO THE OUTER LOOP.  NOW SET UP FOR NEXT LINE.  LOOP IF NOT DONE.
3497                                    ;-
3498  015020  013702  015044            16$:    MOV     70$,R2          ;SET UP TO ROTATE THE DATA PATTERN.
3499  015024  004737  015416                  JSR     PC,ROLDAP       ;ROTATE THE DATA PATTERN.
3500  015030  005205                            INC     R5              ;COUNT THIS LINE
3501  015032  020527  000010                    CMP     R5,#NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
3502  015036  002704                          BLT     4$              ;LOOP IF SOME LINES NOT DONE.
3503
3504  015040                            60$:    PASS                    ;RESTORE GPRS.
3505  015040  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3506  015042  000207                          RTS     PC
3507
```

```
  3508  015044  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 THEM MOV, +2 CLEAR THEN MOV).
                        ;*              ERRNBR - SET UP WITH INITIAL ERROR NUMBER.
                        ;*
                        ;* OUTPUTS:     GPRS0 - GPR SAVE AREA 0 IS DESTROYED.
                        ;*              DEVICE UNDER TEST REGISTERS ARE WRITTEN.
                        ;*              ERROR MESSAGES MAY BE PRINTED AT THE OPERATORS CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,REGTST
                        ;*
                        ;* COMMENTS:    THIS ROUTINE LOOP 16 TIMES WRITING THE SAME DATA PATTERN
                        ;*              ROTATED LEFT ONCE EACH ITERATION.
                        ;*              THIS ROUTINE CAN REPORT ERRORS INITIAL ERRNBR THRU INITIAL+2.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: RDPDR,RCLDAP,SWAP0,WDPDR
                        ;-- ************************************************************************

015046                  REGTST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
015046  004537  002674                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; SET UP THE GPRS FOR THE WRITTING OF THE DATA PATTERN.
                        ;-
015052  012705  000020          MOV     #16.,R5         ;SET UP LOOP COUNTER TO COUNT 16 ITERATIONS.
015056  012702  167410          MOV     #167410,R2      ;INITIALIZE THE DATA PATTERN.
015062  032704  000001          BIT     #BIT0,R4        ;TEST FOR R/W ACCESS.
015066  001001                  BNE     2$              ;R/M/W ACCESS?  YES, R4 IS ALL SET UP.
015070  005004                  CLR     R4              ;NO, INDICATE R/W ACCESS.
015072                  2$:
                        ;+
                        ; SET UP THE GPRS FOR THE CLEARING OR SETTING OF ALL THE USED BITS.
                        ;-
015072  010400                  MOV     R4,R0           ;PASS OPERATION TYPE INDICATOR AROUND SWAP0.
015074  004737  016156          JSR     PC,SWAP0        ;GET ALTERNATE GPR SET IN R1 THRU R5.
015100  013701  002666          MOV     ERRNBR,R1       ;SAVE THE INITIAL ERROR NUMBER.
015104  010004                  MOV     R0,R4
015106  005404                  NEG     R4              ;SET UP OP TYPE FOR CLEARING OR SETTING.
015110  005002                  CLR     R2              ;SET UP CLEAR WRITE PATTERN.
015112  026627  000012  000002  CMP     R4SLOT(SP),#2   ;TEST FOR CLEAR THEN MOV TEST SEQUENCE.
015120  001401                  BEQ     4$              ;CLEAR THEN MOV?  YES, LEAVE WRITE PAT CLEAR.
015122  005102                  COM     R2              ;NO, SET ALL BITS OF WRITE PATTERN.
015124  005003          4$:     CLR     R3              ;INDICATE THAT WORD ACCESSES SHOULD BE USED.
015126  005000                  CLR     R0              ;SET ALTERNATE BYTE EXPECTED DATA PAT TO CLEAR.
015130  026627  000012  177776  CMP     R4SLOT(SP),#-2  ;TEST FOR SET THEN MOV TEST SEQUENCE.
015136  001001                  BNE     6$              ;SET THEN MOV?  YES, LEAVE ALT BYTE PAT CLEAR.
015140  005100                  COM     R0              ;NO, SET ALT BYTE EXPECTED DATA PAT TO ALL 1'S.
015142  004737  016156          6$:     JSR     PC,SWAP0        ;RESTORE SWAPPED GPR VALUES TO R1 THRU R5.
                        ;+
```

```
3565                                    ; START OF DATA PATTERN LOOP.
3566                                    ;-
3567  015146                            8$:
3568                                        ;+
3569                                        ; SET OR CLEAR ALL THE USED BITS OF THE DEVICE REGISTERS FOR ALL LINES.
3570                                        ; VERIFY THAT ALL THE BITS WERE SET OR CLEARED CORRECTLY.
3571                                        ;-
3572  015146  004737  016156                   JSR     PC,SWAP0        ;GET ALTERNATE GPRS FOR SETTING INTIAL STATES.
3573  015152  004737  016550                   JSR     PC,WDPDR        ;GO CLEAR ALL USED REGISTER BITS, ALL LINES.
3574  015156  010137  002666                    MOV     R1,ERRNBR       ;SET UP ERROR NUMBER TO INITIAL ERRNBR.
3575  015162  004737  014626                   JSR     PC,RDPDR        ;VERIFY ALL USED REGISTER BITS, ALL LINES.
3576  015166  004737  016156                   JSR     PC,SWAP0        ;RESTORE MAIN GPRS CONTENTS.
3577                                        ;+
3578                                        ; WRITE DATA PATTERNS, ALL LOWER BYTE USED BITS, ALL REGISTERS, ALL LINES.
3579                                        ; VERIFY THAT THE DATA PATTERN WAS WRITTEN CORRECTLY.
3580                                        ;-
3581  015172  004737  016550                   JSR     PC,WDPDR        ;WRITE DATA PATTERN TO DEVICE REGISTERS.
3582  015176  005237  002666                    INC     ERRNBR          ;SET ERROR NUMBER TO INITIAL+1.
3583  015202  004737  014626                   JSR     PC,RDPDR        ;VERIFY DATA PATTERN IN ALTERRED BYTE(S).
3584  015206  005703                            TST     R3              ;CHECK THE BYTE INDICATOR.
3585  015210  001411                           BEQ     10$             ;WORD ACCESS?  YES, SKIP SECOND BYTE CHECK.
3586                                        ;+
3587                                        ; CHECK THAT THE ALTERNATE (UNMODIFIED) BYTE IS CLEAR OR SET AS EXPECTED.
3588                                        ;-
3589  015212  010201                            MOV     R2,R1           ;SAVE THE DATA PATTERN.
3590  015214  010002                            MOV     R0,R2           ;GET THE ALTERNATE BYTE EXPECTED DATA.
3591  015216  005403                            NEG     R3              ;INDICATE THAT OTHER BYTE IS TO BE CHECKED.
3592  015220  005237  002666                    INC     ERRNBR          ;SET ERROR NUMBER TO INITIAL+2.
3593  015224  004737  014626                   JSR     PC,RDPDR        ;VERIFY DATA PATS IN OTHER BYTES OF REGISTERS.
3594  015230  005403                            NEG     R3              ;RESTORE BYTE INDICATOR.
3595  015232  010102                            MOV     R1,R2           ;RESTORE DATA PATTERN.
3596                                        ;+
3597                                        ; PEPARE THE NEXT DATA PATTERN AND LOOP IF NOT DONE.
3598                                        ;-
3599  015234  004737  015416        10$:        JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
3600  015240  005305                            DEC     R5              ;COUNT THIS ITERATION OF THE LOOP.
3601  015242  003341                           BGT     8$              ;ALL PATTERNS DONE?  NO, LOOP.
3602                                                                    ;YES, RESTORE ERROR NUMBER AND EXIT.
3603  015244  013737  002406  002666  60$:     MOV     GPRS0B,ERRNBR   ;GET THE ERROR NUMBR FROM GPR SWAP STORAGE.
3604  015252                                    PASS                    ;RESTORE GPRS.
3605  015252  004736                                        JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3606  015254  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:
                        ;-- ************************************************************************

015256                  REPSMR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
015256  004537  002674                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015262  005737  002420          TST     ERSMRF                  ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
015266  001404                  BEQ     60$                     ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
                        ;+
                        ; WE HAVE SOME ERROR SUMMARIES TO REPORT.
                        ;-
015270  012737  013130  002672          MOV     #ER9004,ERRBLK  ;SELECT ERROR REPORTING ROUTINE.
                            ;+
                            ; REPORT
                            ;   "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
                            ;-
015276                      ERROR
015276  104460                                                  TRAP    C$ERROR

015300  004736          60$:    PASS                            ;RESTORE GPRS.
015300                                  JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
015302  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.
                        ;*****************************************************************************

015304                  RESETT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
015304  004537  002674                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015310  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.
                        ;-
015314  013704  002214          MOV     CSRA,R4         ;GET THE ADDRESS OF THE DUT'S CSR.
015320  030214          BIT     R2,(R4)         ;CHECK STATE OF MASTER RESET BIT.
015322  001406          BEQ     2$              ;DON'T DELAY IF MR IS ALREADY CLEAR.
015324  005003          CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
015326  012701  004704          MOV     #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
015332  004737  014166          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
015336  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.
                        ;-
015340  010277  164650  2$:     MOV     R2,@CSRA        ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
015344  004737  016100          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.
                          ;-
015350  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
015352  012701  004704          MOV     #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
015356  004737  014166          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
```

```
3707  015362  103410                  BCS     6$              ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
3708                                  ;+
3709                                  ; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET,TEST ABORTED".
3710                                  ; INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
3711                                  ;-
3712  015364  012701  010705         4$:     MOV     #EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
3713  015370  012737  013050  002672         MOV     #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
3714                                          ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
3715                                          ; "TEST ABORTED"
3716  015376                                  ERROR                   ;               >>>>> ERROR <<<<<
3717  015376  104460                                                                    TRAP    C$ERROR
3718  015400  000241                          CLC                     ;INDICATE TEST IS TO BE ABORTED.
3719  015402  000403                          BR      60$             ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
3720                                  ;+
3721                                  ; CLEAR TX AND RX INTERRUPT ENABLE STATUS FLAGS IN IESTAT.
3722                                  ; EXIT WITH CONTINUE TEST INDICATOR SET (IE,CARRY SET).
3723                                  ;-
3724  015404  005037  002276         6$:      CLR    IESTAT          ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
3725  015410  000261                   SEC                           ;INDICATE SUCCESS, CONTINUE TEST.
3726
3727  015412                          60$:    PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
3728  015412  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3729                                                                  ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
3730  015414  000207                          RTS     PC
3731
```

```
3732                                    .SBTTL  GLOBAL SUBROUTINE                - ROLDAP -
3733                                    ;*************************************************************************
3734                                    ;*      - ROTATE LEFT DATA PATTERN
3735                                    ;*      THIS ROUTINE ROTATES THE PASSED INPUT DATA PATTERN LEFT,WITHOUT GOING
3736                                    ;*      THROUGH THE CARRY.THE CARRY IS INITIALLY SET OR CLEARED DEPENDING
3737                                    ;*      UPON THE STATE OF THE MSB OF THE DATA PATTERN,BEFORE A ROL INSTRUCTION
3738                                    ;*      IS EXECUTED.
3739                                    ;*
3740                                    ;* INPUTS:      R2 - CONTAINS THE DATA PATTERN TO BE ROTATED
3741                                    ;*
3742                                    ;* OUTPUTS:     R2 - CONTAINS THE ROTATED DATA PATTERN
3743                                    ;*
3744                                    ;* CALLING SEQUENCE:    JSR     PC,ROLDAP
3745                                    ;*
3746                                    ;* COMMENTS:
3747                                    ;*
3748                                    ;* SUBORDINATE ROUTINES CALLED: NONE
3749                                    ;*************************************************************************
3750
3751  015416                            ROLDAP::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
3752  015416  004537  002674                   JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3753  015422  010202                    MOV     R2,R2                   ;SET PROCESSOR STATUS CODES
3754  015424  005702                    TST     R2                      ;CHECK MSB
3755  015426  100402                    BMI     2$                      ;BRANCH IF SET
3756  015430  000241                    CLC                             ;CLEAR CARRY BIT IF MSB CLEAR
3757  015432  000401                    BR      4$                      ;
3758  015434  000261                2$: SEC                             ;SET CARRY IF MSB SET
3759  015436  006102                4$: ROL     R2                      ;ROTATE DATA PATTERN LEFT
3760  015440                        60$:PASS    R2                      ;RESTORE GPRS,EXCEPT
3761  015440  010266  000006                  MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
3762  015444  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3763                                                                    ;R2 - CONTAINS THE ROTATED DATA PATTERN
3764  015446  000207                    RTS     PC
```

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

015450                  RSTRPT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
015450  004537  002674                  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.
                        ;-
015454  005003                      CLR     R3              ;CLEAR THE CODE COUNTER.
015456  013705  002666              MOV     ERRNBR,R5       ;SAVE ERRNBR FOR RESTORATION LATER.
015462  017702  164530      2$:     MOV     @RBUFA,R2       ;READ A CHAR FROM THE DUT FIFO.
015466  100412                  BMI     4$                  ;SKIP ERROR IF DATA.VALID SET FOR CHAR.
                            ;+
                            ; WE EXPECT A SELFTEST CODE, BUT THIS FIFO SLOT IS EMPTY.
                            ;-
015470  010537  002666              MOV     R5,ERRNBR       ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
015474  012701  011630              MOV     #EM9018,R1      ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
015500  012737  013232  002672      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."
                            ;-
015506                      ERROR                           ;               >>>>> ERROR <<<<<.
015506  104460                                                                       TRAP    C$ERROR
                            ;+
                            ; INIDICATE "SUCCESS" (BECAUSE FIFO IS PURGED), AND EXIT THIS ROUTINE.
                            ;-
015510  000261                  SEC                         ;SET SUCCESS FLAG.
015512  000545                  BR      60$                 ;EXIT ROUTINE.
```

```
3821                                    ;+
3822                                    ; DETERMINE IF THIS IS NOT A SELFTEST CODE.
3823                                    ;-
3824  015514  012700  070001      4$:     MOV     #70001,R0       ;GENERATE BIT MAP OF ANY CLEAR ERROR BITS OR
3825  015520  040200                      BIC     R2,R0           ;  BIT 0 WHICH ARE CLEAR.
3826  015522  001033                    BNE     8$              ;GO TO REPORT ERROR IF THIS IS NOT A TEST CODE.
3827                                    ;+
3828                                    ; WE HAVE A TEST CODE (EITHER BMP OR SELFTEST CODE).
3829                                    ; DETERMINE WHAT TYPE OF CODE WE HAVE.
3830                                    ;-
3831  015524  032702  000200               BIT     #BIT7,R2        ;TEST ROM VERSION CODE INDICATOR BIT.
3832  015530  001443                    BEQ     10$             ;SKIP ERRORS IF SELFTEST ROM VERSION CODE.
3833  015532  120227  000203             CMPB    R2,#203         ;CHECK IF SKIP SELF TEST CODE.
3834  015536  001440                    BEQ     10$             ;SKIP ERROR REPORT IF SKIP SELF TEST CODE FOUND
3835  015540  120227  000201             CMPB    R2,#201         ;CHECK IF NULL CODE PRESENT.
3836  015544  001435                    BEQ     10$             ;SKIP ERROR REPORT IF SELF TEST NULL CODE.
3837  015546  012700  000300             MOV     #300,R0         ;TEST CODE TYPE BITS FOR BOTH CODE
3838  015552  040200                      BIC     R2,R0           ;  TYPE BITS SET (INDICATING BMP CODE).
3839  015554  001003                    BNE     6$              ;IF IT IS NOT A BMP CODE GO REPORT ERROR.
3840  015556  004737  016032             JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
3841  015562  000426                    BR      10$             ;GO GET THE NEXT CHARACTER FROM THE FIFO.
3842                                    ;+
3843                                    ; WE HAVE A SELFTEST ERROR CODE.
3844                                    ;-
3845  015564  010537  002666      6$:     MOV     R5,ERRNBR       ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
3846  015570  005237  002666             INC     ERRNBR          ;CALCULATE INITIAL ERROR NUMBER PLUS 1.
3847  015574  012701  011655             MOV     #EM9020,R1      ;PASS ERROR MESSAGE INFO TO ER9008 ROUTINE.
3848  015600  012737  013310  002672       MOV     #ER9008,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3849                                    ;+
3850                                    ; REPORT ERROR WITH NUMBER INITIAL ERRNRB + 1.
3851                                    ;   'UNEXPECTED SELFTEST ERROR CODE FOR LINE NN IN FIFO AFTER RESET:"
3852                                    ;-
3853  015606                            ERROR                   ;               >>>>> ERROR <<<<<.
3854  015606  104460                                                                    TRAP    C$ERROR
3855  015610  000413                    BR      10$             ;GO TO END OF LOOP.
3856                                    ;+
3857                                    ; WE HAVE A NON-SELFTEST CODE (EITHER BMP CODE OR DATA CHAR).
3858                                    ;-
3859  015612  010537  002666      8$:     MOV     R5,ERRNBR       ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
3860  015616  062737  000002  002666       ADD     #2,ERRNBR       ;CALCULATE INITIAL ERROR NUMBER PLUS 2.
3861  015624  012701  011640             MOV     #EM9019,R1      ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
3862  015630  012737  013232  002672       MOV     #ER9007,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3863                                    ;+
3864                                    ; REPORT ERROR WITH NUMBER INITIAL ERRNRB + 2.
3865                                    ;   'NON-SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET."
3866                                    ;-
3867  015636                            ERROR                   ;               >>>>> ERROR <<<<<.
3868  015636  104460                                                                    TRAP    C$ERROR
3869                                    ;+
3870                                    ; END OF LOOP, LOOP IF NOT ALL CHARS HAVE BEEN READ FROM THE FIFO.
3871                                    ;-
3872  015640  005203             10$:    INC     R3              ;SET CODE COUNTER FOR NEXT ITERATION OF LOOP.
3873  015642  020327  000010             CMP     R3,#NUMLNS      ;TEST FOR ALL CODES READ.
3874  015646  002705                    BLT     2$              ;LOOP IF NOT CHARS READ FROM FIFO.
3875                                    ;+
3876                                    ; PURGE THE FIFO UNTIL DATA.VALID IS CLEAR OR UNTIL TOO MANY CHARS ARE READ.
```

```
3877                                    ;-
3878  015650  012704  000022                    MOV     #18.,R4         ;INITIALIZE THE CHARACTER COUNTER.
3879  015654  010537  002666                    MOV     R5,ERRNBR       ;GET INITIAL VALUE OF THE ERROR NUMBER.
3880  015660  062737  000003  002666            ADD     #3,ERRNBR       ;CALCULATE ERROR NUMBER OF NEXT ERROR.
3881  015666  012737  013310  002672            MOV     #ER9008,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3882  015674  017702  164316              12$:  MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE DUT FIFO.
3883  015700  000261                        SEC                     ;INDICATE SUCCESS IN CASE DATA.VALID IS CLEAR.
3884  015702  100051                        BPL     60$             ;EXIT ROUTINE WITH SUCCESS IF DATA.VALID CLEAR.
3885                                  ;+
3886                                  ; WE HAVE A CHARACTER.
3887                                  ; DETERMINE IF CHARACTER IS A DATA CHARACTER.
3888                                  ;-
3889  015704  012700  070000                    MOV     #70000,R0       ;TEST BITS 12 THRU 14 OF THE
3890  015710  040200                        BIC     R2,R0           ;  CODE READ FROM THE DUT FIFO.
3891  015712  001403                        BEQ     14$             ;SKIP THIS ERROR IF CODE IS NOT A DATA CHAR.
3892                                  ;+
3893                                  ; WE HAVE AN UNEXPECTED DATA CHARACTER: SET UP AND GO TO REPORT ERROR.
3894                                  ;-
3895  015714  012701  011701                    MOV     #EM9022,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3896  015720  000423                        BR      22$             ;GO TO REPORT THIS ERROR.
3897                                  ;+
3898                                  ; WE HAVE AN UNEXPECTED CODE.
3899                                  ; DETERMINE IF THE CODE IS A MODEM STATUS CODE.
3900                                  ;-
3901  015722  032702  000001              14$:  BIT     #BIT0,R2        ;TEST MODEM STATUS INDICATOR BIT OF CODE.
3902  015726  001003                        BNE     16$             ;SKIP THIS ERROR IF NOT MODEM STATUS CODE.
3903                                  ;+
3904                                  ; WE HAVE A MODEM STATUS CODE: SET UP AND GO TO REPORT ERROR.
3905                                  ;-
3906  015730  012701  011720                    MOV     #EM9023,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3907  015734  000415                        BR      22$             ;GO TO REPORT THIS ERROR.
3908                                  ;+
3909                                  ; WE HAVE AN ONBOARD TEST CODE.
3910                                  ; DETERMINE IF THIS CODE IS A BMP CODE.
3911                                  ;-
3912  015736  032702  000200              16$:  BIT     #BIT7,R2        ;TEST THE ROM VERSION BIT OF THE CODE.
3913  015742  001404                        BEQ     18$             ;GOTO SET UP FOR SELFTEST CODE IF ROM VERSION.
3914  015744  012700  000300                    MOV     #300,R0
3915  015750  040200                        BIC     R2,R0           ;TEST THE ERROR TYPE BITS OF THE CODE.
3916  015752  001403                        BEQ     20$             ;SKIP THIS ERROR IF BMP CODE.
3917                                  ;+
3918                                  ; WE HAVE A SELFTEST CODE: SET UP AND GO TO REPORT ERROR.
3919                                  ;-
3920  015754  012701  011742              18$:  MOV     #EM9024,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3921  015760  000403                        BR      22$             ;GO TO REPORT THIS ERROR.
3922                                  ;+
3923                                  ; WE HAVE A BMP CODE: SAVE IT ON THE QUEUE.
3924                                  ;-
3925  015762  004737  016032              20$:  JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
3926  015766  000401                        BR      24$             ;
3927                                  ;+
3928                                  ; REPORT THE ERROR WITH ERROR NUMBER OF INITIAL ERRNBR + 3.
3929                                  ;   'UNEXPECTED XXX XXXX FOR LINE NN IN FIFO AFTER RESET:"
3930                                  ;-
3931  015770                          22$:  ERROR                   ;               >>>>> ERROR <<<<<.
3932  015770  104460                                                                  TRAP    C$ERROR
```

```
3933                                    ;+
3934                                    ; END OF LOOP.
3935                                    ; COUNT THE CHARACTER WE JUST RECEIVED, AND CHECK FOR TOO MANY RECEIVED.
3936                                    ;-
3937  015772  005304                24$:    DEC     R4              ;COUNT THIS CHARACTER.
3938  015774  001337                        BNE     12$             ;LOOP IF NOT TOO MANY CHARACTERS PURGED.
3939                                  ;+
3940                                  ; WE READ TOO MANY VALID CHARACTERS WHILE TRYING TO PURGE THE FIFO.
3941                                  ; REPORT ERROR AND EXIT WITHOUT SUCCESS.
3942                                  ;   "FIFO WILL NOT PURGE (DATA.VALID STUCK SET), REMAINDER OF TEST SKIPPED."
3943                                  ;-
3944  015776  012701  011517                MOV     #EM9017,R1      ;SELECT PROPER ERROR MESSAGE.
3945  016002  010537  002666                MOV     R5,ERRNBR       ;GET INITIAL ERROR NUMBER.
3946  016006  062737  000004  002666        ADD     #4,ERRNBR       ;CALCULATE INITIAL ERRNBR + 4.
3947  016014  012737  012576  002672        MOV     #ER0503,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3948                                        ;PRINT ERROR REPORT.
3949  016022                                ERROR                   ;               >>>>> ERROR <<<<<.
3950  016022  104460                                                                  TRAP    C$ERROR
3951  016024  000241                        CLC                     ;CLEAR THE SUCCESS FLAG.
3952
3953  016026                        60$:    PASS                    ;RESTORE GPRS,
3954  016026  004736                                JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3955  016030  000207                        RTS     PC              ; CARRY - SUCCESS FLAG (SET IF FIFO IS PURGED).
```

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

016032                  SAVBMP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016032  004537  002674                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016036  013704  002462          MOV     BMPCQP,R4               ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
016042  113724  002274          MOVB    TSTNUM,(R4)+            ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
016046  005204                  INC     R4                      ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
016050  042702  177400          BIC     #177400,R2              ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
016054  010224                  MOV     R2,(R4)+                ;SAVE THE BMP CODE ON THE QUEUE.
016056  020427  002664          CMP     R4,#BMPCQE              ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
016062  103402                 BLO     2$                      ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
016064  162704  000004          SUB     #4,R4                   ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
016070  010437  002462  2$:     MOV     R4,BMPCQP               ;SAVE THE POINTER.

016074                  60$:    PASS                            ;RESTORE GPRS.
016074  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
016076  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.
                        ;-- ************************************************************************

016100                  SKPSTS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016100  004537  002674                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016104  012704  000012          MOV     #10.,R4         ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
016110  004737  014126          JSR     PC,DELAY        ;DELAY FOR 10 MILLI-SECONDS.
                        ;+
                        ; WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
                        ;-
016114  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.
016120  012703  052525          MOV     #52525,R3       ;INITIALISE THE SKIP SELF-TEST CODE.
016124  005301          4$:     DEC     R1              ;SELECT THE NEXT SET OF DEVICE REGISTERS.
016126  013704  002214          MOV     CSRA,R4         ;GET THE ADDRESS OF THE CSR OF THE DUT.
016132  010124                  MOV     R1,(R4)+        ;SELECT A BANK OF DUT REGISTERS.
016134  010324          6$:     MOV     R3,(R4)+        ;WRITE THE CODE TO A DUT REGISTER.
016136  020437  002232          CMP     R4,TXBFCA       ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
016142  103774                BLO     6$              ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
016144  032701  000017          BIT     #17,R1          ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
016150  001365                BNE     4$              ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.

016152                  60$:    PASS                            ;RESTORE GPRS.
016152  004736                        JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
016154  000207                RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - SWAP0 -
                                ;++ ************************************************************************************
                                ;*                          - 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,SWAP0
                                ;*
                                ;* COMMENTS:    THE STATE OF THE CARRY FLAG IS NOT ALTERRED BY THIS ROUTINE.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************************

016156  010046                  SWAP0::  MOV    R0,-(SP)        ;SAVE THE CONTENTS OF R0.
                                ;+
                                ; LOAD THE STACK FROM THE GPRS.
                                ;-
016160  010146                          MOV     R1,-(SP)        ;SAVE THE CONTENTS OF R1.
016162  010246                          MOV     R2,-(SP)        ;SAVE THE CONTENTS OF R2.
016164  010346                          MOV     R3,-(SP)        ;SAVE THE CONTENTS OF R3.
016166  010446                          MOV     R4,-(SP)        ;SAVE THE CONTENTS OF R4.
016170  010546                          MOV     R5,-(SP)        ;SAVE THE CONTENTS OF R5.
                                ;+
                                ; LOAD THE GPRS FROM THE GPR SAVE AREA 0.
                                ;-
016172  012700  002406                  MOV     #GPRS0B,R0      ;GET THE BASE ADDRESS OF GPR SAVE AREA 0.
016176  012001                          MOV     (R0)+,R1        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 1.
016200  012002                          MOV     (R0)+,R2        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 2.
016202  012003                          MOV     (R0)+,R3        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 3.
016204  012004                          MOV     (R0)+,R4        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 4.
016206  012005                          MOV     (R0)+,R5        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 5.
                                ;+
                                ; LOAD THE GPR SAVE AREA 0 FROM THE STACK.
                                ;-
016210  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 5 WITH SAVED R5.
016212  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 4 WITH SAVED R4.
016214  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 3 WITH SAVED R3.
016216  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 2 WITH SAVED R2.
016220  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 1 WITH SAVED R1.

016222  012600                          MOV     (SP)+,R0        ;RESTORE THE INITIAL VALUE OF R0.

016224  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.
                                ;-- ************************************************************************

016226                          TSABRT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016226  004537  002674                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016232  012701  016250                 MOV     #2$,R1                  ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
016236  012737  013050  002672         MOV     #ER1603,ERRBLK          ;SET-UP THE ERROR REPORTING ROUTINE.
016244                                  ERROR                           ;                   >>>>> ERROR <<<<<.
016244  104460                                                                                TRAP    C$ERROR
016246  000432                         BR      60$                     ;
016250  047040  047117  051055  2$:    .ASCIZ / NON-RELATED TEST ERROR FOUND DURING TEST EXECUTION/
016256  046105  052101  042105
016264  052040  051505  020124
016272  051105  047522  020122
016300  047506  047125  020104
016306  052504  044522  043516
016314  052040  051505  020124
016322  054105  041505  052125
016330  047511  000116
                                .EVEN
016334                          60$:    PASS                            ;RESTORE GPRS.
016334  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
016336  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - UNSDIV -
                        ;++ ************************************************************************
                        ;*                      - UNSIGNED DIVIDE ROUTINE -
                        ;*      THIS SUBROUTINE IS USED TO DIVIDE A 32 BIT UNSIGNED DIVIDEND BY A
                        ;*      16 BIT UNSIGNED DIVISOR GIVING A 16 BIT QUOTIENT.  ALL NUMBERS ARE
                        ;*      CONSIDERED TO BE UNSIGNED.  A SUCCESS FLAG IS NOT SET ON RETURN IF
                        ;*      THE QUOTIENT WAS TOO BIG TO BE CONTAINED IN 16 BITS.
                        ;*
                        ;* INPUTS:      R1 - THE DIVISOR, UNSIGNED, 16 BITS.
                        ;*              R2 - MOST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
                        ;*              R3 - LEAST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
                        ;*
                        ;* OUTPUTS:     R1 - QUOTIENT, UNSIGNED, 16 BITS (177777 IF OVERFLOW).
                        ;*              CARRY - SUCCESS FLAG, SET IF COMPLETE QUOTIENT FITS IN 16 BITS.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,UNSDIV
                        ;*
                        ;* COMMENTS:    IF THE DIVISOR IS 0 THE QUOTIENT IS RETURNED AS ALL ONES
                        ;*              (177777) AND THE CARRY IS CLEAR REGARDLESS OF THE DIVIDEND.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

016340                  UNSDIV:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016340  004537  002674                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
                        ;-
016344  010204                  MOV     R2,R4           ;GET MSW OF DIVIDEND FOR SUBTRACT.
016346  160104                  SUB     R1,R4           ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
016350  103403                  BCS     2$              ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
016352  012701  177777          MOV     #-1,R1          ;SET QUOTIENT TO ALL ONES (177777),
016356  000442                  BR      60$             ;EXIT WITH CARRY CLEAR.
                        ;+
                        ; SET UP COUNTERS AND VARIOUS WORKING GPRS.
                        ;-
016360  005004          2$:     CLR     R4              ;CLEAR THE LSW OF THE DIVISOR.
016362  000241                  CLC                     ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
016364  006001                  ROR     R1              ; DIVISOR BY
016366  006004                  ROR     R4              ; 2(UNSIGNED)
016370  012700  000020          MOV     #16.,R0         ;SET UP INITIAL SHIFT COUNT TO 16.
                        ;+
                        ; THE SUBTRACT AND SHIFT LOOP.
                        ;-
016374  010246          4$:     MOV     R2,-(SP)        ;SAVE MSWORD OF DIVIDEND.
016376  010346                  MOV     R3,-(SP)        ;SAVE LSWORD OF DIVIDEND.
016400  160403                  SUB     R4,R3           ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
016402  005602                  SBC     R2              ;MSWORD DIVIDEND - BORROW .
016404  103402                  BCS     6$              ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
016406  160102                  SUB     R1,R2           ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
016410  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.
                            ;-
016412  012603          6$:     MOV     (SP)+,R3        ;RESTORE LSWORD OF DIVIDEND.
```

```
4186  016414  012602                  MOV     (SP)+,R2        ;RESTORE MSWORD OF DIVIDEND.
4187  016416  000401                  BR      10$             ;GOTO SHIFT 1 INTO THE QUOTIENT.
4188                                ;+
4189                                ; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
4190                                ; COMPLEMENTED LATER).   CARRY IS CLEAR.
4191                                ;-
4192  016420  012626            8$:     MOV     (SP)+,(SP)+     ;POP THE SAVED DIVIDEND OFF OF THE STACK.
4193                                ;+
4194                                ; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.
4195                                ;-
4196  016422  006105            10$:    ROL     R5              ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
4197  016424  000241                    CLC                     ;DIVIDE THE
4198  016426  006001                    ROR     R1              ;  DEVISOR BY
4199  016430  006004                    ROR     R4              ;  2 (UNSIGNED).
4200  016432  005300                    DEC     R0              ;COUNT THIS SHIFT AND SUBTRACT.
4201  016434  001357                    BNE     4$              ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
4202  016436  005105                    COM     R5              ;GET QUOTIENT FROM INVERTED QUOTIENT.
4203                                ;+
4204                                ; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.
4205                                ;-
4206  016440  000241                    CLC                     ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
4207  016442  006103                    ROL     R3              ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
4208  016444  103402                    BCS     12$             ;IF CARRY FROM SHIFT, ROUND UP.
4209  016446  160403                    SUB     R4,R3           ;SUBTRACT DIVISOR FROM DIVIDEND.
4210  016450  103403                    BCS     14$             ;IF BORROW, DON'T ROUND UP.
4211                                ;+
4212                                ; ROUND UP, EXTRA SUBTRACT WENT.
4213                                ;-
4214  016452  005205            12$:    INC     R5              ;INCREMENT THE QUOTIENT BY ONE.
4215  016454  001001                    BNE     14$             ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
4216  016456  005305                    DEC     R5              ;DON'T LET ROUNDING CAUSE OVERFLOW.
4217                                ;+
4218                                ; ALL DONE, PASS QUOTIENT AND EXIT.
4219                                ;-
4220  016460  010501            14$:    MOV     R5,R1           ;PASS QUOTIENT BACK IN R1.
4221  016462  000261                    SEC                     ;INDICATE NO OVERFLOW.
4222
4223  016464                     60$:    PASS    R1                      ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
4224  016464  010166  000004                                MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
4225  016470  004736                                       JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
4226                                                                 ;R1 - 16 BIT, UNSIGNED QUOTIENT,
4227  016472  000207                    RTS     PC              ;CARRY - SET INDICATES NO OVERFLOW (SUCCESS).
```

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

016474                  WAIBIS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016474  004537  002674                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016500  010204                  MOV     R2,R4                   ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
016502  010102                  MOV     R1,R2
016504  042701  170000          BIC     #170000,R1              ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
016510  042702  007777          BIC     #7777,R2                ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
016514  000302                  SWAB    R2                      ;PUT LINE NUMBER FIELD IN LSBYTE.
016516  006202                  ASR     R2                      ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
016520  006202                  ASR     R2                      ;  POSITION TO USE IT AS A WORD TABLE OFFSET
016522  006202                  ASR     R2                      ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
016524  016202  002346          MOV     BITTBL(R2),R2           ;GET BIT MAP OF LINE TO TEST FROM TABLE.
016530  010203                  MOV     R2,R3                   ;INDICATE THAT THE BIT SHOULD BE SET.
016532  004737  014166          JSR     PC,MSLGET               ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
                                                                ;  CARRY IS CORRECT UPON MSLGET RETURN.
016536  010002                  MOV     R0,R2                   ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
016540                  60$:    PASS    R2                      ;RESTORE GPRS, EXCEPT THE FOLLOWING:
016540  010266  000006                          MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
016544  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
016546  000207                  RTS     PC                      ;  CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).
```

```
4276                            .SBTTL  GLOBAL SUBROUTINE                - WDPDR -
4277                            ;++ ******************************************************************************
4278                            ;*                      - WRITE DATA PATTERN TO DEVICE REGISTERS -
4279                            ;*      THIS ROUTINE WRITES A ROTATED DATA PATTERN TO EACH OF THE 6 DEVICE
4280                            ;*      REGISTERS OF EACH ACTIVE LINE OF THE DEVICE UNDER TEST.
4281                            ;*      THE DATA PATTERN IS ROTATED ONCE AFTER EACH WRITE TO A DEVICE REGISTER
4282                            ;*              ON A PARTICULAR LINE.  THE STARTING DATA PATTERN FOR EACH LINE
4283                            ;*              IS ROTATED ONCE AFTER WRITING ALL THE REGISTERS ON A PARTICULAR
4284                            ;*              LINE.  THIS LEADS TO THE FOLLOWING DATA PATTERN:
4285                            ;*                      LINE 0, REGISTER 0 - SHIFTED 0 BIT POSITIONS
4286                            ;*                      LINE 0, REGISTER 1 - SHIFTED 1 BIT POSITION
4287                            ;*                      ...
4288                            ;*                      LINE 1, REGISTER 0 - SHIFTED 1 BIT POSITION
4289                            ;*                      LINE 2, REGISTER 1 - SHIFTED 2 BIT POSITIONS
4290                            ;*                      ...
4291                            ;*      ANY BITS FIELDS IN THE DEVICE REGISTERS THAT CANNOT BE ALTERED
4292                            ;*      ARE MASKED OUT OF THE DATA PATTERN BEFORE IT IS WRITTEN.
4293                            ;*      THIS ROUTINE WILL USE EITHER MOV, MOVB, BIS, BISB, BIC, OR BICB
4294                            ;*      INSTRUCTIONS.  THE UPPER OR LOWER BYTE CAN BE SPECIFIED FOR WRITING.
4295                            ;*
4296                            ;* INPUTS:      R2 - USED TO PASS IN THE DATA PATTERN TO BE ROTATED & WRITTEN.
4297                            ;*              R3 - BYTE INDICATOR (- => LO BYTE, + => HI BYTE, 0 => BOTH).
4298                            ;*              R4 - OPERATION TYPE INDICATOR (- => BIC, + => BIS, 0 => MOV).
4299                            ;*              ACTLNS - BIT MAP OF THE ACTIVE LINES ON THE DEVICE UNDER TEST.
4300                            ;*              CSRA - CONTAINS THE CSR ADDRESS OF THE DEVICE UNDER TEST.
4301                            ;*              DRADRT - BASE ADDRESS OF DEVICE REGISTER ADDRESS TABLE.
4302                            ;*              LPRO - EQUATED TO LPR REG OFFSET FROM DEVICE CSR ADDRESS.
4303                            ;*              NUMLNS - NUMBER OF LINES ON THE DEVICE UNDER TEST.
4304                            ;*              TXBFCO - EQUATED TO TBUFFCT REG OFFSET FROM DEVICE CSR ADDRESS.
4305                            ;*              UNBTTB - BASE ADDRESS OF THE UNUSED BIT TABLE.
4306                            ;*
4307                            ;* OUTPUTS:     DEVICE REGISTERS ON ALL ACTIVE DEVICE LINES ARE MODIFIED.
4308                            ;*
4309                            ;* CALLING SEQUENCE:    JSR     PC,WDPDR
4310                            ;*
4311                            ;* COMMENTS:    THIS ROUTINE DOES NOT WRITE ANY DATA TO THE TX.CHAR REGISTERS.
4312                            ;*              THE CSR IS CLEARED EXCEPT FOR THE IND.ADR.REG FIELD.
4313                            ;*
4314                            ;* SUBORDINATE ROUTINES CALLED: ROLDAP.
4315                            ;-- ******************************************************************************
4316
4317  016550                    WDPDR:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
4318  016550  004537  002674                    JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
4319                            ;+
4320                            ; SET UP OUTER LOOP WHICH WRITES THE DATA PATTERN TO EACH LINE'S REGISTERS
4321                            ;-
4322  016554  005005                    CLR     R5                      ;CLEAR LINE COUNTER TO SELECT LINE 0.
4323                            ;+
4324                            ; THE OUTER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP WRITES DATA TO ALL OF
4325                            ; THE DEVICE REGISTERS FOR A PARTICULAR LINE IF IT IS ACTIVE.
4326                            ;-
4327  016556  010204            2$:     MOV     R2,R4                   ;SAVE THE OUTER LOOP DATA PATTERN.
4328  016560  010577  163430            MOV     R5,@CSRA                ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
4329  016564  006305                    ASL     R5                      ;TURN LINE NUMBER INTO A WORD OFFSET.
4330  016566  036537  002346  002210    BIT     BITTBL(R5),ACTLNS
4331  016574  001451                    BEQ     20$                     ;LINE ACTIVE?  NO, SKIP THIS LINE.
```

```
 4332  016576  012701  000004                    MOV     #LPRO,R1        ;YES, INITIALIZE THE REGISTER OFFSET.
 4333                                        ;+
 4334                                        ;       THE INNER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP WRITES DATA TO A
 4335                                        ;       DEVICE REGISTER.
 4336                                        ;-
 4337  016602  010200                    4$:     MOV     R2,R0
 4338  016604  046100  002234                    BIC     UNBTTB(R1),R0   ;CLEAR BIT FIELDS FOR UNUSED REGISTER BITS.
 4339  016610  016103  002214                    MOV     DRADRT(R1),R3   ;GET THE ADDRESS OF THE DEVICE REGISTER.
 4340  016614  005766  000010                    TST     R3SLOT(SP)      ;CHECK THE OPERAND TYPE INDICATOR.
 4341  016620  003402                            BLE     6$              ;HIGH BYTE?  NO, SKIP HIGH BYTE ADDRESS SET UP.
 4342  016622  005203                            INC     R3              ;YES, SET THE REG ADDRESS TO THE HIGH BYTE.
 4343  016624  000300                            SWAB    R0              ;MOVE HIGH BYTE DATA INTO THE LOW BYTE.
 4344  016626  005766  000010            6$:     TST     R3SLOT(SP)      ;CHECK THE OPERAND TYPE INDICATOR.
 4345  016632  001412                            BEQ     12$             ;WORD ACCESS?  YES, GO PERFORM WORD ACCESS.
 4346                                        ;+
 4347                                        ;PERFORM BYTE ACCESS TO THE SPECIFIED BYTE OF THE SPECIFIED REGISTER.
 4348                                        ;-
 4349  016634  005766  000012                    TST     R4SLOT(SP)      ;NO, CHECK THE ACCESS TYPE INDICATOR.
 4350  016640  100403                            BMI     8$              ;USE BIC?  YES, GO PERFORM BICB INSTRUCTION.
 4351  016642  001404                            BEQ     10$             ;USE MOV?  YES, GO PERFORM MOVB INSTRUCTION.
 4352  016644  150013                            BISB    R0,(R3)         ;NEITHER.  PERFORM BISB ACCESS TO REGISTER.
 4353  016646  000415                            BR      18$
 4354  016650  140013            8$:             BICB    R0,(R3)         ;PERFORM BICB ACCESS TO REGISTER.
 4355  016652  000413                            BR      18$
 4356  016654  110013            10$:            MOVB    R0,(R3)         ;PERFORM MOVB ACCESS TO REGISTER.
 4357  016656  000411                            BR      18$
 4358                                        ;+
 4359                                        ;PERFORM WORD ACCESS TO THE SPECIFIED REGISTER.
 4360                                        ;-
 4361  016660  005766  000012            12$:    TST     R4SLOT(SP)      ;CHECK THE ACCESS TYPE INDICATOR.
 4362  016664  100403                            BMI     14$             ;USE BIC?  YES, GO PERFORM BIC INSTRUCTION.
 4363  016666  001404                            BEQ     16$             ;USE MOV?  YES, GO PERFORM MOV INSTRUCTION.
 4364  016670  050013                            BIS     R0,(R3)         ;NEITHER.  PERFORM BIS ACCESS TO REGISTER.
 4365  016672  000403                            BR      18$
 4366  016674  040013            14$:            BIC     R0,(R3)         ;PERFORM BIC ACCESS TO REGISTER.
 4367  016676  000401                            BR      18$
 4368  016700  010013            16$:            MOV     R0,(R3)         ;PERFORM MOV ACCESS TO REGISTER.
 4369                                        ;+
 4370                                        ; PREPARE THE DATA PATTERN AND OFFSET FOR THE NEXT REGISTER ON THIS LINE.
 4371                                        ;-
 4372  016702  004737  015416            18$:    JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
 4373  016706  062701  000002                    ADD     #2,R1           ;INCREMENT OFFSET FOR NEXT REGISTER.
 4374  016712  020127  000016                    CMP     R1,#TXBFCO      ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
 4375  016716  003731                            BLE     4$              ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
 4376                                        ;+
 4377                                        ; BACK INTO THE OUTER LOOP.  NOW SET UP FOR NEXT LINE.  LOOP IF NOT DONE.
 4378                                        ;-
 4379  016720  010402            20$:            MOV     R4,R2           ;SET UP TO ROTATE THE DATA PATTERN.
 4380  016722  004737  015416                    JSR     PC,ROLDAP       ;ROTATE THE DATA PATTERN.
 4381  016726  006205                            ASR     R5              ;CONVERT BACK TO LINE NUMBER FROM WORD OFFSET.
 4382  016730  005205                            INC     R5              ;COUNT THIS LINE.
 4383  016732  020527  000010                    CMP     R5,#NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
 4384  016736  002707                            BLT     2$              ;LOOP IF SOME LINES NOT DONE.
 4385
 4386  016740                            60$:    PASS                    ;RESTORE GPRS.
 4387  016740  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
```

```
 4388  016742  000207                          RTS     PC
```

```
                                .SBTTL  INTERRUPT SERVICE ROUTINE       - CLKINT -
                                ;++ ************************************************************************
                                ;*      THIS ROUTINE IS EXECUTED CLKHRZ TIMES PER SECOND.  IT DECREMENTS THE
                                ;*      TWO TIMER COUNTERS DOWN TO ZERO.
                                ;*
                                ;* INPUTS:      TIMER1 - TIMER COUNTER #1.
                                ;*              TIMER2 - TIMER COUNTER #2.
                                ;*              TIMER3 - TIMER COUNTER FOR CALL OF BREAK MACRO.
                                ;*
                                ;* OUTPUTS:     THE 2 TIMER COUNTERS ARE DECREMENTED IF THEY ARE NOT ZERO.
                                ;*
                                ;* CALLING SEQUENCE:    PUT #CLKINT IN THE CLOCK INTERRUPT VECTOR SLOT.
                                ;*                      PUT THE DESIRED TIME PERIOD (SECONDS TIMES CLKHRZ) IN
                                ;*                      EITHER TIMER1 OR TIMER2 AND POLL THE RESPECTIVE TIMER
                                ;*                      COUNTER TO DETECT ITS GOING TO 0 ON TIME-OUT.
                                ;*
                                ;* COMMENTS:    THE 2 COUNTERS WILL NOT WRAPAROUND BUT WILL STOP AT 0.  THIS
                                ;*              ALLOWS THE DETECTION OF A TIME-OUT ANY TIME AFTER THE TIME-OUT
                                ;*              HAS OCCURRED UNTIL THE TIMER COUNTER IS SET TO ANOTHER VALUE.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

016744  005737  002322        CLKINT:: TST     TIMER1          ;CHECK FOR TIMER1 AT ZERO.
016750  001402                         BEQ     2$              ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
016752  005337  002322                 DEC     TIMER1          ;DECREMENT TIME COUNT.
016756  005737  002324        2$:     TST     TIMER2          ;CHECK FOR TIMER2 AT ZERO.
016762  001402                         BEQ     4$              ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
016764  005337  002324                 DEC     TIMER2          ;DECREMENT TIME COUNT.
016770  005337  002326        4$:     DEC     TIMER3          ;DECREMENT THE BREAK COUNT.
016774  001006                         BNE     60$             ;EXIT IF NOT TIME TO CALL BREAK.
016776  013737  002330  002326        MOV     BCOUNT,TIMER3   ;SET UP TIME TILL NEXT BREAK.
017004  010046                         MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 FROM BREAK MACRO.
017006                                 BREAK                   ;CHECK FOR OPERATOR CONTROL/C.
017006  104422                                                         TRAP    C$BRK
017010  012600                         MOV     (SP)+,R0        ;RESTORE CONTENTS OF R0.
017012  000002                60$:     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.
                                ;*************************************************************************

017014  021627  014010          TP4RTN:: CMP    (SP),#ADRPTR    ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
017020  001402                          BEQ     2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
017022  000177  163254                  JMP     @TP4VEC         ;IF NOT,JUMP TO NORMAL 004 TRAP SERVICE RTN.
017026  052737  100000  002304  2$:     BIS     #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
017034  000002                          RTI                     ;ALL DONE, GO BACK TO THE TEST.
```

```
4455
4456
4457                            .SBTTL  REPORT CODING SECTION
4458
4459                            ;++
4460                            ; THE REPORT CODING SECTION CONTAINS THE
4461                            ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
4462                            ;--
4463
4464  017036                            BGNRPT
4465  017036                                                          L$RPT::
4466
4467  017036                            EXIT    RPT
4468  017036  000167                                                         .WORD   J$JMP
4469  017040  000000                                                         .WORD   L10014-2-.
4470
4471                            .EVEN
4472
4473  017042                            ENDRPT
4474  017042                                                          L10014:
4475  017042  104425                                                         TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE


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

017044                                  BGNPROT
017044                                                                  L$PROT::

017044  177777                          -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
017046  177777                          -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
017050  177777                          -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

017052                                  ENDPROT
```

```
                                ;++
                                ;*****************************************************************************
                                ;*      THIS SECTION CONTAINS THE CODE WHICH IS PERFORMED AT THE BEGINNING OF
                                ;*      EACH PASS OR AFTER A CONTINUE COMMAND.
                                ;*      THIS CODE PERFORMS THE FOLLOWING ACTIONS:
                                ;*
                                ;*      MOVES THE INFORMATION HELD IN THE HARDWARE P-TABLE INTO THE GLOBAL
                                ;*      DATA AREA.
                                ;*
                                ;*****************************************************************************
                                ;--
017052                                  BGNINIT
017052                                                                  L$INIT::

                                ;SEE IF PROGRAM JUST STARTED, BR IF YES
017052                                  READEF  #EF.START
017052  012700  000040                                                  MOV     #EF.START,R0
017056  104447                                                          TRAP    C$REFG
017060                                  BCOMPLETE       NEWSTA
017060  103416                                                          BCS     NEWSTA

                                ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
017062                                  READEF  #EF.RESTART
017062  012700  000037                                                  MOV     #EF.RESTART,R0
017066  104447                                                          TRAP    C$REFG
017070                                  BCOMPLETE       NEWRES
017070  103555                                                          BCS     NEWRES

                                ;SEE IF THIS IS A NEW PASS, BR IF YES
017072                                  READEF  #EF.NEW
017072  012700  000035                                                  MOV     #EF.NEW,R0
017076  104447                                                          TRAP    C$REFG
017100                                  BCOMPLETE       NEWPAS
017100  103554                                                          BCS     NEWPAS

                                ;SEE IF PROGRAM WAS JUST CONTINUED
017102                                  READEF  #EF.CONTINUE
017102  012700  000036                                                  MOV     #EF.CONTINUE,R0
017106  104447                                                          TRAP    C$REFG
017110                                  BNCOMPLETE      GETPRM
017110  103160                                                          BCC     GETPRM
017112  000137  017632                 JMP     ENDIT
017116                          NEWSTA:
017116                                  BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
017116  104433                                                          TRAP    C$RESET

                                ;+
                                ; SET UP FOR LINE TIME CLOCK INTERRUPTS.
                                ;-
017120                                  CLOCK   L,R1            ;GET THE CLOCK PARAMETERS.
017120  012700  000114                                                  MOV     #'L,R0
017124  104462                                                          TRAP    C$CLCK
017126  010001                                                          MOV     R0,R1
017130  012137  002312                  MOV     (R1)+,CLKCSR            ;STORE CLOCK CSR ADDRESS.
017134  012137  002314                  MOV     (R1)+,CLKBRL            ;STORE CLOCK BUS REQ INT LEVEL.
017140  012137  002316                  MOV     (R1)+,CLKVEC            ;STORE CLOCK INTERRUPT VECTOR.
017144  012137  002320                  MOV     (R1)+,CLKHRZ            ;STORE CLOCK FREQUENCY.
017150  023727  002320  000062          CMP     CLKHRZ,#50.             ;TEST FOR 50HZ LINE FREQUENCY.
```

```
4549  017156  001004                          BNE     2$                      ;BRANCH IF CLOCK IS NOT 50HZ.
4550  017160  012737  000024  002332          MOV     #20.,MSTICK             ;INDICATE 20MS PER CLOCK TICK.
4551  017166  000403                          BR      4$
4552  017170  012737  000021  002332  2$:     MOV     #17.,MSTICK             ;INDICATE 17 MS PER CLOCK TICK.
4553  017176                          4$:     SETVEC  CLKVEC,#CLKINT,PRI06    ;INITIALIZE CLOCK INTERRUPT VECTOR.
4554  017176  013746  000300                                                          MOV     PRI06,-(SP)
4555  017202  012746  016744                                                          MOV     #CLKINT,-(SP)
4556  017206  013746  002316                                                          MOV     CLKVEC,-(SP)
4557  017212  012746  000003                                                          MOV     #3,-(SP)
4558  017216  104437                                                                 TRAP    C$SVEC
4559  017220  062706  000010                                                         ADD     #10,SP
4560  017224  013700  002320                  MOV     CLKHRZ,R0               ;INITIALIZE THE BREAK COUNT
4561  017230  006300                          ASL     R0                      ; TO CAUSE A BREAK
4562  017232  010037  002330                  MOV     R0,BCOUNT               ; EVERY 2 SECONDS.
4563  017236                                  SETPRI  #PRI05                  ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
4564  017236  012700  000240                                                          MOV     #PRI05,R0
4565  017242  104441                                                                 TRAP    C$SPRI
4566                                  ;+
4567                                  ; ENABLE THE LINE TIME CLOCK (LTC) CHECKING TO MAKE SURE THAT THE CSR
4568                                  ; IS ACCESSABLE.
4569                                  ;    FIRST SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
4570                                  ;-
4571  017244  013737  000004  002302          MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
4572  017252  012737  017014  000004          MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4573                                  ;+
4574                                  ;    ENABLE LTC CHECKING FOR 004 TRAP IN CASE CSR IS NOT THERE.
4575                                  ;-
4576  017260  005037  002304                  CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
4577  017264  012737  000100  002306          MOV     #BIT6,WORD1     ;SET UP TO SET BIT6 OF THE LTC CSR.
4578  017272  012700  002306                  MOV     #WORD1,R0       ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
4579  017276  013701  002312                  MOV     CLKCSR,R1       ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
4580  017302  004737  013776                  JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
4581  017306  013737  002302  000004          MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4582  017314  103403                          BCS     6$              ;IF NO TRAP, LTC IS THERE SO CONTINUE.
4583  017316  005037  002320                  CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
4584  017322  000402                          BR      8$              ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
4585                                  ;+
4586                                  ; CALIBRATE THE DELAY ROUTINE MILLI-SECOND DELAY COUNT VALUE.
4587                                  ;-
4588  017324  004737  013552          6$:     JSR     PC,CALMSL
4589                                  ;+
4590                                  ; CHECK FOR MEMMORY MANAGEMENT PRESENT ON THIS MACHINE.
4591                                  ; IF MEM MGT IS PRESENT, DISABLE IT.
4592                                  ;-
4593  017330  013737  000004  002302  8$:     MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
4594  017336  012737  017014  000004          MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4595  017344  005037  002304                  CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
4596  017350  005037  002306                  CLR     WORD1           ;PREPARE TO CLEAR THE MEM MGT SR0 REGISTER.
4597  017354  012700  002306                  MOV     #WORD1,R0       ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
4598  017360  013701  002336                  MOV     MMSR0,R1        ;SELECT MEM MGT SR0 REGISTER AS DESTINATION.
4599  017364  005037  002340                  CLR     MMPRES          ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
4600  017370  005037  002342                  CLR     MMENAB          ;INDICATE MEM MGT IS NOT ENABLED.
4601  017374  004737  013776                  JSR     PC,CKTRAP       ;CLEAR THE MEM MCT SR0 REG AND CHECK FOR TRAP.
4602  017400  013737  002302  000004          MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4603  017406  103003                          BCC     10$             ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
4604  017410  012737  000001  002340          MOV     #1,MMPRES       ;INDICATE THAT MEM MGT IS PRESENT.
```

```
4605  017416  005037  002300          10$:    CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4606  017422  000403                          BR      NEWPAS
4607
4608  017424                          NEWRES: BRESET                  ;AVOID ILLEGAL INTERRUPT PROBLEMS.
4609  017424  104433                                                                  TRAP    C$RESET
4610  017426  005037  002300                  CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4611  017432                          NEWPAS:
4612  017432  012737  177777  002212          MOV     #-1,UNITN       ;RESET LOGICAL DEVICE TO -1
4613                                  ;+
4614                                  ; INCREMENT THE PASS COUNTER, CORRECT FOR ANY OVERFLOW.
4615                                  ;  THIS COUNTER IS USED IN THE ROM VERSION TEST.
4616                                  ;-
4617  017440  005237  002300                  INC     PASCNT          ;INCREMENT THE PASS COUNTER.
4618  017444  001002                          BNE     GETPRM          ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
4619  017446  005337  002300                  DEC     PASCNT          ;SET PASS COUNT TO 177777 OCTAL.
4620
4621                                  ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
4622  017452                          GETPRM:
4623  017452  005237  002212                  INC     UNITN           ;INCREMENT LOGICAL DEVICE NUMBER
4624  017456  023737  002212  002012          CMP     UNITN,L$UNIT    ;SEE IF MAXIMUM UNIT NO. EXCEEDED
4625  017464  002362                          BGE     NEWPAS          ;BR IF YES
4626
4627  017466                                  GPHARD  UNITN,R1        ;GET P-TABLE POINTER INTO R1
4628  017466  013700  002212                                                          MOV     UNITN,R0
4629  017472  104442                                                                  TRAP    C$GPHRD
4630  017474  010001                                                                  MOV     R0,R1
4631  017476                                  BCOMPLETE       30$     ;BR IF DEVICE AVAILABLE
4632  017476  103401                                                                  BCS     30$
4633  017500  000764                          BR      GETPRM          ;SKIP THIS DEVICE
4634
4635
4636                                          ;************ HARDWARE PARAMETER MOVING CODE ****************
4637  017502  012137  002214          30$:    MOV     (R1)+,CSRA      ;STORE DHV-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
4638  017506  012137  002210                  MOV     (R1)+,ACTLNS    ;STORE THE ACTIVE LINES BIT MAP.
4639                                  ;+
4640                                  ;       CALCULTATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
4641                                  ;       DEVICE REGISTER ADDRESS TABLE.
4642                                  ;-
4643  017512  013701  002214                  MOV     CSRA,R1         ;COPY CSR ADDRESS
4644  017516  005201                          INC     R1              ;INCREMENT CSR ADDRESS
4645  017520  005201                          INC     R1              ; COPY BY 2.
4646  017522  012703  000007                  MOV     #7,R3           ;SET UP REGISTER COUNT
4647  017526  012702  002216                  MOV     #RBUFA,R2       ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
4648  017532  010122          12$:    MOV     R1,(R2)+        ;STORE REGISTER ADDRESS IN TABLE
4649  017534  005201                          INC     R1              ;INCREMENT REGISTER ADDRESS
4650  017536  005201                          INC     R1              ; BY 2,FOR THE NEXT DEVICE REGISTER.
4651  017540  005303                          DEC     R3              ;DECREMENT REGISTER COUNT
4652  017542  001373                          BNE     12$             ;LOOP IF NOT DONE
4653                                  ;+
4654                                  ; INITIALISE THE BMP CODE QUEUE.
4655                                  ;-
4656  017544  012700  002464                  MOV     #BMPCQB,R0      ;GET THE START ADDRESS OF THE QUEUE.
4657  017550  012701  002664                  MOV     #BMPCQE,R1      ;GET THE END ADDRESS OF THE QUEUE.
4658  017554  010037  002462                  MOV     R0,BMPCQP       ;SET THE POINTER TO THE START OF THE QUEUE.
4659  017560  005020          14$:    CLR     (R0)+           ;CLEAR OUT THE CONTENTS OF THE QUEUE.
4660  017562  020001                          CMP     R0,R1           ;CHECK IF END OF QUEUE HAS BEEN REACHED.
```

```
4661  017564  103775                          BLO     14$             ;LOOP IF NOT ALL DONE.
4662                                          ;+
4663                                          ; REPORT THE UNIT NUMBER IF THE SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
4664                                          ;  AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
4665                                          ;-
4666  017566  032737  000020  002204          BIT     #BIT4,OPTION    ;CHECK IF THE QUESTION WAS ANSWERED YES.
4667  017574  001416                          BEQ     16$             ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
4668  017576  023727  002012  000001          CMP     L$UNIT,#1       ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
4669  017604  003412                          BLE     16$             ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
4670  017606                                  PRINTF  #MFUNIT,UNITN   ;REPORT UNIT NUMBER.
4671  017606  013746  002212                                                  MOV     UNITN,-(SP)
4672  017612  012746  002772                                                  MOV     #MFUNIT,-(SP)
4673  017616  012746  000002                                                  MOV     #2,-(SP)
4674  017622  010600                                                          MOV     SP,R0
4675  017624  104417                                                          TRAP    C$PNTF
4676  017626  062706  000006                                                  ADD     #6,SP
4677  017632                          16$:
4678
4679  017632  005037  002310          ENDIT:  CLR     CTRLCF          ;CLR THE CTRL-C TEST ABORT FLAG.
4680                                          ;+
4681                                          ; SET THE PROCESSOR PRIORITY TO ALLOW LTC INTERRUPTS BUT NOT OTHERS.
4682                                          ;-
4683  017636                                  SETPRI  #PRI07                  ;SET PROCESSOR PRIORITY TO 5.
4684  017636  012700  000340                                                  MOV     #PRI07,R0
4685  017642  104441                                                          TRAP    C$SPRI
4686
4687  017644                                  ENDINIT
4688  017644                                                          L10016:
4689  017644  104411                                                          TRAP    C$INIT
```

```
                                .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.
                                ;--

017646                                  BGNAUTO
017646                                                          L$AUTO::


017646                                  ENDAUTO
017646                                                          L10017:
017646  104461                                                          TRAP    C$AUTO
```

```
4709
4710
4711                                    .SBTTL  CLEANUP CODING SECTION
4712
4713                                    ;++
4714                                    ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4715                                    ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
4716                                    ;--
4717
4718  017650                                    BGNCLN
4719  017650                                                                          L$CLEAN::
4720
4721  017650  005737  002310                     TST     CTRLCF          ;DID WE GET HERE BY CTRL-C FROM TEST?
4722  017654  001401                            BEQ     2$              ;CTRL-C FROM TEST?  NO, SKIP BUS RESET.
4723  017656                                    BRESET                  ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
4724  017656  104433                                                                  TRAP    C$RESET
4725  017660                                2$:
4726  017660                                    EXIT    CLN
4727  017660  104432                                                                  TRAP    C$EXIT
4728  017662  000002                                                                  .WORD   L10020-.
4729
4730
4731                                    .EVEN
4732
4733  017664                                    ENDCLN
4734  017664                                                                  L10020:
4735  017664  104412                                                                  TRAP    C$CLEAN
```

```
4736
4737
4738                                    .SBTTL  DROP UNIT SECTION
4739
4740                                    ;++
4741                                    ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4742                                    ; TO NO LONGER BE TESTED.
4743                                    ;--
4744
4745  017666                                    BGNDU
4746  017666                                                                          L$DU::
4747  017666                                     PRINTF #DROP,R0        ;REPORT UNIT THAT HAS BEEN DROPPED.
4748  017666  010046                                                                  MOV     R0,-(SP)
4749  017670  012746  017712                                                         MOV     #DROP,-(SP)
4750  017674  012746  000002                                                         MOV     #2,-(SP)
4751  017700  010600                                                                 MOV     SP,R0
4752  017702  104417                                                                 TRAP    C$PNTF
4753  017704  062706  000006                                                         ADD     #6,SP
4754  017710  000427                            BR      EDROP           ;BRANCH AROUND THE MESSAGE.
4755
4756  017712  040445  052440  044516  DROP:    .ASCIZ/%A UNIT%D6%A DROPPED FROM FURTHER TESTING.%N/
4757  017720  022524  033104  040445
4758  017726  042040  047522  050120
4759  017734  042105  043040  047522
4760  017742  020115  052506  052122
4761  017750  042510  020122  042524
4762  017756  052123  047111  027107
4763  017764  047045     000
4764          017770                    .EVEN
4765  017770                    EDROP:
4766  017770                            EXIT    DU
4767  017770  000167                                                              .WORD   J$JMP
4768  017772  000000                                                              .WORD   L10021-2-.
4769
4770
4771
4772  017774                            ENDDU
4773  017774                                                          L10021:
4774  017774  104453                                                          TRAP    C$DU
```

```
4775
4776
4777                                    .SBTTL  ADD UNIT SECTION
4778
4779                                    ;++
4780                                    ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
4781                                    ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
4782                                    ;  TO THE TEST CYCLE.
4783                                    ;--
4784
4785  017776                                    BGNAU
4786  017776                                                                  L$AU::
4787  017776                                    EXIT    AU
4788  017776  000167                                                                .WORD   J$JMP
4789  020000  000000                                                                .WORD   L10022-2-.
4790
4791
4792                                    .EVEN
4793
4794  020002                                    ENDAU
4795  020002                                                                  L10022:
4796  020002  104452                                                                TRAP    C$AU
4797
```

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

020004                                  BGNTST
020004                                                                  T1::
        000001                          TNUM == 1               ;THIS TEST MUST ALWAYS BE INCLUDED AS TEST 1.
020004  012737  000001  002274           MOV    #TNUM,TSTNUM     ;SET THE TEST NUMBER TO 1.
020012  012737  177777  002310           MOV    #-1,CTRLCF       ;INDICATE THAT WE ARE IN A TEST.
                                ;+
                                ; SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
                                ;-
020020  013737  000004  002302           MOV    4,TP4VEC         ;SAVE THE EXISTING 004 TRAP VECTOR.
020026  012737  017014  000004           MOV    #TP4RTN,4        ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
020034  005005                           CLR    R5               ;CLEAR THE ERROR FLAGS.

                                ;+
                                ; SET UP FOR THE INITIAL ITERATION OF THE TEST LOOP:
                                ;-
020036  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.
                                ;-
020040  005037  002304          2$:      CLR    TP4FLG           ;CLEAR THE 004 TRAP FLAG.
020044  013700  002214                   MOV    CSRA,R0          ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
020050  012701  020264                   MOV    #52$,R1          ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
020054  004737  013776                   JSR    PC,CKTRAP        ;MOVE AND CHECK FOR TRAP.
020060  103402                           BCS    4$               ;IF NO TRAP, BYPASS ERROR.
020062  052705  100001                   BIS    #100001,R5       ;SET FATAL READ ERROR FLAGS.
020066  042737  000017  020264  4$:      BIC    #17,52$          ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
020074  050437  020264                   BIS    R4,52$           ;OR IN THE LINE COUNTER TO THE I.A.R FIELD.
020100  010100                           MOV    R1,R0            ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
020102  013701  002214                   MOV    CSRA,R1          ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
020106  004737  013776                   JSR    PC,CKTRAP        ;MOVE AND CHECK FOR TRAP.
020112  103403                           BCS    6$               ;IF NO TRAP, BYPASS ERROR.
020114  052705  100002                   BIS    #100002,R5       ;SET FATAL WRITE ERROR FLAGS.
020120  000440                           BR     40$              ;EXIT AND REPORT FATAL ERROR.

                                ;+
                                ;    NOW, WE TEST EACH REGISTER FOR THIS LINE.
                                ;-
020122  012702  000010          6$:      MOV    #10,R2           ;INIT REGISTER COUNTER TO 8.
020126  013737  002214  020262           MOV    CSRA,50$                 ;INITIALIZE THE REGISTER POINTER.
020134  012700  020262          8$:      MOV    #50$,R0          ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
020140  012701  020264                   MOV    #52$,R1          ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
```

```
 4854  020144  004737  013776              JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
 4855  020150  103402                      BCS     10$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
 4856  020152  052705  100001               BIS     #100001,R5      ;SET FATAL READ ERROR FLAGS.
 4857  020156  010100              10$:     MOV     R1,R0           ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
 4858  020160  012701  020262               MOV     #50$,R1         ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.
 4859  020164  004737  013776              JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
 4860  020170  103402                      BCS     12$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
 4861  020172  052705  100002               BIS     #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
 4862  020176  005237  020262      12$:     INC     50$             ;INCREMENT THE REGISTER
 4863  020202  005237  020262               INC     50$             ; POINTER BY 2.
 4864  020206  005302                       DEC     R2              ;COUNT THE REGISTER.
 4865  020210  001351                      BNE     8$              ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
 4866
 4867                                  ;+
 4868                                  ; NOW WE SET UP TO TEST THE NEXT LINE, OR TO EXIT IF WE ARE DONE.
 4869                                  ;-
 4870  020212  005204                       INC     R4              ;INCREMENT THE LINE COUNTER.
 4871  020214  020427  000010               CMP     R4,#NUMLNS      ;COMPARE LINE COUNTER AGAINST NUMBER OF LINES.
 4872  020220  002707                      BLT     2$              ;LOOP TO TEST THE NEXT LINE IF WE'RE NOT DONE.
 4873
 4874                                  ;+
 4875                                  ; DONE CHECKING DEVICE REGISTER ADDRESSES.
 4876                                  ;    REPORT ANY ERRORS AND EXIT.
 4877                                  ;-
 4878  020222  013737  002302  000004  40$:      MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
 4879  020230  005705                       TST     R5              ;CHECK THE ERROR FLAGS.
 4880  020232  100015                      BPL     60$             ;EXIT ROUTINE IF NO ERRORS.
 4881                                      ; REPORT "DEVICE REGISTER ACCESS ERRORS"
 4882  020234                               ERRDF   101,EM0103,ER0101;              >>>>> ERROR #101 <<<<<.
 4883  020234  104455                                                                  TRAP    C$ERDF
 4884  020236  000145                                                                  .WORD   101
 4885  020240  004572                                                                  .WORD   EM0103
 4886  020242  012176                                                                  .WORD   ER0101
 4887
 4888  020244                              DODU    UNITN           ;DROP THIS UNIT FROM FUTHER TESTING.
 4889  020244  013700  002212                                                         MOV     UNITN,R0
 4890  020250  104451                                                                 TRAP    C$DODU
 4891  020252  005037  002310               CLR     CTRLCF          ;INDICATE NO CTRL-C ABORT FROM TEST.
 4892  020256                              DOCLN                   ;ABORT THIS SUB PASS.
 4893  020256  104444                                                                 TRAP    C$DCLN
 4894  020260  000402                      BR      60$             ;
 4895                                  ;+
 4896                                  ; LOCAL STORAGE.
 4897                                  ;-
 4898  020262  000000              50$:    .WORD   0       ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 4899  020264  000000              52$:    .WORD   0       ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 4900  020266  005037  002310      60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 4901  020272                              ENDTST
 4902  020272                                                                  L10023:
 4903  020272  104401                                                                 TRAP    C$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.
020274  012737  000002  002274          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (2)
020302  012737  177777  002310          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
020310                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
020310  012700  000240                                                          MOV     #PRI05,R0
020314  104441                                                                  TRAP    C$SPRI
020316  012737  000001  002664          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
020324  012737  004630  002670          MOV     #EM0201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
020332  012737  012514  002672          MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;-
020340  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
020344  012702  000040                  MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
020350  005003                          CLR     R3              ;WAITING FOR BIT TO CLEAR.
020352  013704  002214                  MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
020356  004737  014166                  JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
020362  103410                          BCS     2$              ;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.
                                ;-
020364                                  BRESET                  ;NO, TRY TO JOG DEVICE WITH BUS RESET.
020364  104433                                                                  TRAP    C$RESET
020366  004737  016100                  JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
020372  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
020376  004737  014166                  JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
020402  103016                          BCC     4$              ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
                                ;+
                                ; SET THE MASTER RESET BIT AND VERIFY THAT IT CLEARS WITHIN THE PROPER TIME.
                                ;-
020404  012701  011610          2$:     MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
020410  010214                          MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
020412  004737  014166                  JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
020416  103010                          BCC     4$              ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
020420  012702  011610                  MOV     #5000.,R2
020424  160102                          SUB     R1,R2           ;CALCULATE # OF MS FOR MR TO CLEAR.
020426  001413                          BEQ     6$              ;GO REPORT ERROR IF MR CLEAR IMMEDIATELY.
020430  020227  000764                  CMP     R2,#500.
020434  002417                          BLT     8$              ;GO REPORT ERROR IF MR CLEAR IN < 1/2 SECOND.
020436  000424                          BR      60$             ;EXIT THE TEST WITHOUT ERROR.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
020440  012737  000311  002666  4$:      MOV     #201.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
020446  012701  004676                  MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
```

```
4960  020452                          ERROR                   ;REPORT ERROR.          >>>>> ERROR #201 <<<<<
4961  020452  104460                                                                          TRAP    C$ERROR
4962  020454  000415                  BR      60$             ;EXIT THE TEST.
4963
4964                          ;REPORT MR BIT CLEAR IMMEDIATELY AFTER DUT RESET.
4965  020456  012737  000312  002666  6$:     MOV     #202.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
4966  020464  012701  005051          MOV     #EM0203,R1      ;SELECT ERROR MESSAGE.
4967  020470                          ERROR                   ;REPORT ERROR.          >>>>> ERROR #202 <<<<<
4968  020470  104460                                                                          TRAP    C$ERROR
4969  020472  000406                  BR      60$             ;EXIT THE TEST.
4970
4971                          ;REPORT MR CLEAR WITHIN 1/2 SECOND OF DUT RESET.
4972  020474  012737  000313  002666  8$:     MOV     #203.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
4973  020502  012701  005214          MOV     #EM0204,R1      ;SELECT ERROR MESSAGE.
4974  020506                          ERROR                   ;REPORT ERROR.          >>>>> ERROR #203 <<<<<
4975  020506  104460                                                                          TRAP    C$ERROR
4976
4977  020510                  60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
4978  020510  012700  000340                                                                  MOV     #PRI07,R0
4979  020514  104441                                                                          TRAP    C$SPRI
4980  020516  005037  002310          CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
4981  020522                          ENDTST
4982  020522                                                                          L10024:
4983  020522  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.
020524 012737 000003 002274      MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (3)
020532 012737 177777 002310      MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
                                 SETPRI #PRI05          ;ALLOW LTC INTERRUPTS.
020540 012700 000240                                                    MOV     #PRI05,R0
020544 104441                                                           TRAP    C$SPRI
020546 012737 000001 002664      MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
020554 012737 005373 002670      MOV    #EM0301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
020562 012737 012514 002672      MOV    #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;-
020570 012701 011610             MOV    #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
020574 012702 000040             MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
020600 005003                    CLR    R3              ;WAITING FOR BIT TO CLEAR.
020602 013704 002214             MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
020606 004737 014166            JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
020612 103410                    BCS    2$              ;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.
                                ;-
                                        BRESET                  ;NO, TRY TO JOG DEVICE WITH BUS RESET.
020614                                                                  TRAP    C$RESET
020614 104433
020616 004737 016100            JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
020622 012701 011610             MOV    #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
020626 004737 014166            JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
020632 103024                    BCC    6$              ;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.
                                ;-
020634 012701 000310            2$:     MOV    #200.,R1        ;TIME-OUT VALUE IS 1/5 SECOND.
020640 010214                    MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
020642 004737 016100            JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
020646 004737 014166            JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
020652 103007                    BCC    4$              ;GO FIND OUT WHAT IS WRONG IF MR NOT CLEAR.
020654 012702 000310             MOV    #200.,R2
020660 160102                    SUB    R1,R2           ;CALCULATE # OF MS FOR MR TO CLEAR.
020662 020227 000012             CMP    R2,#10.
020666 002415                   BLT     8$              ;GO REPORT ERROR IF MR CLEAR IN < 10 MS.
020670 000431                    BR     60$             ;EXIT THE TEST WITHOUT ERROR.
                                ;+
                                ; MR DID NOT CLEAR WITHIN 1/5 SECOND, SEE IF IT CLEARS WITHIN 5 SECONDS.
                                ;-
020672 012701 011300            4$:     MOV    #4800.,R1       ;TIME-OUT VALUE IS 5 SECONDS MINUS 1/5 SECOND.
020676 004737 014166            JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
```

```
5040  020702  103416                          BCS     10$             ;GO REPORT ERROR IF MR CLEARED FINALLY.
5041                                  ;+
5042                                  ; ERROR REPORTS:
5043                                  ;-
5044                                      ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5045  020704  012737  000455  002666  6$:     MOV     #0301.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5046  020712  012701  004676                  MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
5047  020716                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0301 <<<<<
5048  020716  104460                                                                                  TRAP    C$ERROR
5049  020720  000415                          BR      60$             ;EXIT THE TEST.
5050
5051                                      ;REPORT MR BIT CLEAR WITHIN 10 MS AFTER DUT RESET.
5052  020722  012737  000456  002666  8$:     MOV     #0302.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5053  020730  012701  005436                  MOV     #EM0302,R1      ;SELECT ERROR MESSAGE.
5054  020734                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0302 <<<<<
5055  020734  104460                                                                                  TRAP    C$ERROR
5056  020736  000406                          BR      60$             ;EXIT THE TEST.
5057
5058                                      ;REPORT MR CLEARED BETWEEN 1/5 SECOND AND 5 SECONDS OF DUT RESET.
5059  020740  012737  000457  002666  10$:    MOV     #0303.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5060  020746  012701  005576                  MOV     #EM0303,R1      ;SELECT ERROR MESSAGE.
5061  020752                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0303 <<<<<
5062  020752  104460                                                                                  TRAP    C$ERROR
5063
5064  020754                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
5065  020754  012700  000340                                                                          MOV     #PRI07,R0
5066  020760  104441                                                                                  TRAP    C$SPRI
5067  020762  005037  002310                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5068  020766                                  ENDTST
5069  020766                                                                          L10025:
5070  020766  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.
                                                                                        MOV     #PRI05,R0
                                                                                        TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
                                         MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (4)
                                         MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
                                         MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
                                         MOV    #EM0401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                         MOV    #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
                                         MOV    #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
                                         MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
                                         CLR    R3              ;WAITING FOR BIT TO CLEAR.
                                         MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
                                         MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
                                        JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
                                        JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
                                        BCC     4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; READ 6 CHARACTERS FROM THE DUT AND VERIFY THAT THEY ARE VALID SELFTEST
                                ;  CODES.
                                ;-
                                         MOV    (R4)+,R0        ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
                                         MOV    #6,R1           ;INITIALIZE THE LOOP COUNTER.
                                2$:      MOV    (R4),R2         ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
                                         MOV    R2,R0
                                         BIC    #177476,R0      ;REMOVE ALL DUT BITS SPECIFIC TO SELFTEST CODE.
                                         CMP    R0,#201         ;CHECK THAT BITS 0,6, AND 7 ARE CORRECT.
                                        BNE     6$              ;GO REPORT ERROR IF CODE IS NOT SELFTEST CODE.
                                         DEC    R1              ;COUNT THIS LOOP ITERATION.
                                        BNE     2$              ;LOOP IF NOT ALL LINES DONE.
                                        BR      60$             ;EXIT TEST, NO ERROR FOUND.

                                ;+
                                ; ERROR REPORTS:
                                ;-
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
                                4$:      MOV    #0401.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
                                         MOV    #EM0202,R1      ;SELECT ERROR MESSAGE.
                                        ERROR                   ;REPORT ERROR.          >>>>> ERROR #0401 <<<<<
                                                                                                TRAP    C$ERROR
                                        BR      60$             ;EXIT THE TEST.
```

```
5127                                    ;REPORT IMPROPER CODE FOUND IN DUT RBUF AFTER RESET (SKIP SELFTEST).
5128  021136  012737  000622  002666  6$:   MOV    #0402.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5129  021144  012701  006024                MOV    #EM0402,R1      ;SELECT ERROR MESSAGE.
5130  021150                                ERROR                  ;REPORT ERROR.          >>>>> ERROR #0402 <<<<<
5131  021150  104460                                                                              TRAP    C$ERROR
5132
5133  021152                          60$:  SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
5134  021152  012700  000340                                                                      MOV     #PRI07,R0
5135  021156  104441                                                                             TRAP    C$SPRI
5136  021160  005037  002310                CLR    CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5137  021164                                ENDTST
5138  021164                                                                        L10026:
5139  021164  104401                                                                             TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RXFFDA -
                                ;++ ************************************************************************
                                ;*              - RBUF REGISTER RX FLAG FIELD TEST -
                                ;*      THIS TEST VERIFIES THAT THE FIELD OF 3 FLAG BITS IN THE RBUF READS
                                ;*      AS ALL ONES WHEN THE SELFTEST CODES ARE BEING READ FROM THE DUT
                                ;*      AFTER A BOARD RESET AND SKIP SELFTEST SEQUENCE.
                                ;*
                                ;-- ************************************************************************
021166                                  BGNTST
021166                                                                          T5::
021166                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
021166  012700  000240                                                        MOV     #PRI05,R0
021172  104441                                                                TRAP    C$SPRI
                000005                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
021174  012737  000005  002274           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (5)
021202  012737  177777  002310           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
021210  012737  000001  002664           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
021216  012737  006174  002670           MOV     #EM0501,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
021224  012737  012514  002672           MOV     #ER0201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
021232  012701  011610                   MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
021236  012702  000040                   MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
021242  005003                           CLR     R3             ;WAITING FOR BIT TO CLEAR.
021244  013704  002214                   MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
021250  010214                           MOV     R2,(R4)        ;SET THE DUT MASTER RESET BIT.
021252  004737  016100                   JSR     PC,SKPSTS      ;SKIP THE SELFTEST.
021256  004737  014166                   JSR     PC,MSLGET      ;WAIT FOR DUT CSR MR BIT TO CLEAR.
021262  103013                           BCC     4$             ;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.
                                ;-
021264  012400                           MOV     (R4)+,R0       ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
021266  012701  000010                   MOV     #8.,R1         ;INITIALIZE THE LOOP COUNTER.
021272  011402                 2$:       MOV     (R4),R2        ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
021274  012700  070000                   MOV     #70000,R0
021300  040200                           BIC     R2,R0          ;CALCULATE BIT MAP OF CLEAR RX ERROR FLAGS.
021302  001012                           BNE     6$             ;GO REPORT ERROR IF NOT ALL RX ERROR FLAGS SET.
021304  005301                           DEC     R1             ;COUNT THIS LOOP ITERATION.
021306  001371                           BNE     2$             ;LOOP IF NOT ALL LINES DONE.
021310  000415                           BR      60$            ;EXIT TEST, NO ERROR FOUND.

                                ;+
                                ; ERROR REPORTS:
                                ;-
                                   ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
021312  012737  000765  002666  4$:       MOV     #0501.,ERRNBR  ;SET THE ERROR NUMBER IN ERROR TABLE.
021320  012701  004676                    MOV     #EM0202,R1     ;SELECT ERROR MESSAGE.
021324                                    ERROR                  ;REPORT ERROR.          >>>>> ERROR #0501 <<<<<
021324  104460                                                                          TRAP    C$ERROR
021326  000406                           BR      60$            ;EXIT THE TEST.

                                   ;REPORT ONE OR MORE RX ERROR FLAGS FOUND SET WITH SELFTEST CODE.
```

```
 5196  021330  012737  000766  002666  6$:     MOV    #0502.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
 5197  021336  012701  006242                  MOV    #EM0502,R1      ;SELECT ERROR MESSAGE.
 5198  021342                                  ERROR                  ;REPORT ERROR.          >>>>> ERROR #0502 <<<<<
 5199  021342  104460                                                                          TRAP    C$ERROR
 5200
 5201  021344                          60$:    SETPRI #PRI07          ;DISABLE ALL INTERRUPTS.
 5202  021344  012700  000340                                                                  MOV     #PRI07,R0
 5203  021350  104441                                                                         TRAP    C$SPRI
 5204  021352  005037  002310                  CLR    CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5205  021356                                  ENDTST
 5206  021356                                                                        L10027:
 5207  021356  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.
                                ;*
                                ;-- ************************************************************************
021360                                  BGNTST
021360                                                                          T6::
021360                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
021360  012700  000240                                                          MOV     #PRI05,R0
021364  104441                                                                  TRAP    C$SPRI
        000006                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
021366  012737  000006  002274           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (6)
021374  012737  177777  002310           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
021402  012737  000001  002664           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
021410  012737  006416  002670           MOV     #EM0601,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
021416  012737  012514  002672           MOV     #ER0201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
021424  012701  011610                   MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
021430  012702  000040                   MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
021434  005003                           CLR     R3             ;WAITING FOR BIT TO CLEAR.
021436  013704  002214                   MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
021442  010214                           MOV     R2,(R4)        ;SET THE DUT MASTER RESET BIT.
021444  004737  016100                  JSR     PC,SKPSTS      ;SKIP THE SELFTEST.
021450  004737  014166                  JSR     PC,MSLGET      ;WAIT FOR DUT CSR MR BIT TO CLEAR.
021454  103016                          BCC     4$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; CHECK THAT THE RX DATA AVAILABLE BIT IS SET.
                                ;-
021456  032714  000200                    BIT     #BIT7,(R4)     ;TEST THE DUT RX.DATA.AVAIL BIT.
021462  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.
                                ;-
021464  012705  001130                   MOV     #600.,R5       ;ALLOW READING 600 CHARS BEFORE ERROR.
021470  010403                           MOV     R4,R3
021472  012300                           MOV     (R3)+,R0       ;CALCULATE THE RBUF ADDRESS.
021474  011300                  2$:      MOV     (R3),R0        ;READ A CHARACTER FROM THE RX FIFO.
021476  032714  000200                   BIT     #BIT7,(R4)     ;TEST THE DUT RX.DATA.AVAIL BIT.
021502  001427                          BEQ     60$            ;EXIT TEST WITHOUT ERROR IF RX.DATA.AVAIL CLR.
021504  005305                           DEC     R5             ;COUNT THE CHARACTER JUST READ.
021506  001372                          BNE     2$             ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
021510  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.
021512  012737  001131  002666  4$:      MOV     #0601.,ERRNBR  ;SET THE ERROR NUMBER IN ERROR TABLE.
021520  012701  004676                   MOV     #EM0202,R1     ;SELECT ERROR MESSAGE.
021524                                  ERROR                  ;REPORT ERROR.          >>>>> ERROR #0601 <<<<<
```

```
5264  021524  104460                                                          TRAP    C$ERROR
5265  021526  000415                  BR      60$             ;EXIT THE TEST.
5266
5267                          ;REPORT THAT RX.DATA.AVAIL BIT WAS NOT SET AFTER A RESET COMPLETION.
5268  021530  012737  001132  002666  6$:     MOV     #0602.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5269  021536  012701  006452          MOV     #EM0602,R1      ;SELECT ERROR MESSAGE.
5270  021542                          ERROR                   ;REPORT ERROR.          >>>>> ERROR #0602 <<<<<
5271  021542  104460                                                          TRAP    C$ERROR
5272  021544  000406                  BR      60$             ;EXIT THE TEST.
5273
5274                          ;REPORT THAT RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO.
5275  021546  012737  001133  002666  8$:     MOV     #0603.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5276  021554  012701  006632          MOV     #EM0603,R1      ;SELECT ERROR MESSAGE.
5277  021560                          ERROR                   ;REPORT ERROR.          >>>>> ERROR #0603 <<<<<
5278  021560  104460                                                          TRAP    C$ERROR
5279
5280  021562                  60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
5281  021562  012700  000340                                                  MOV     #PRI07,R0
5282  021566  104441                                                          TRAP    C$SPRI
5283  021570  005037  002310          CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5284  021574                          ENDTST
5285  021574                                                          L10030:
5286  021574  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  #PRI05          ;ALLOW LTC INTERRUPTS.
021576 012700 000240                                                                    MOV     #PRI05,R0
021602 104441                                                                           TRAP    C$SPRI
000007                           TNUM == TNUM + 1               ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
021604 012737 000007 002274             MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (7)
021612 012737 177777 002310             MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
021620 012737 000001 002664             MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
021626 012737 007015 002670             MOV     #EM0701,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
021634 012737 012514 002672             MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
021642 012701 011610                    MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
021646 012702 000040                    MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
021652 005003                           CLR     R3              ;WAITING FOR BIT TO CLEAR.
021654 013704 002214                    MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
021660 010214                           MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
021662 004737 016100                    JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
021666 004737 014166                    JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
021672 103012                           BCC     4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; CHECK THAT THE RX DATA VALID BIT IS SET.
                                ;-
021674 012400                           MOV     (R4)+,R0        ;INCREMENT POINTER TO PNT TO DUT RBUF REG.
021676 005714                           TST     (R4)            ;TEST THE DUT RX.DATA.VALID BIT.
021700 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.
                                ;-
021702 012705 001130                    MOV     #600.,R5        ;ALLOW READING 600 CHARS BEFORE ERROR.
021706 011400                   2$:     MOV     (R4),R0         ;READ A CHARACTER FROM THE RX FIFO.
021710 100027                           BPL     60$             ;EXIT TEST WITHOUT ERROR IF BIT IS CLEAR.
021712 005305                           DEC     R5              ;COUNT THE CHARACTER JUST READ.
021714 001374                           BNE     2$              ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
021716 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.
021720 012737 001275 002666 4$:         MOV     #0701.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
021726 012701 004676                    MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
021732                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0701 <<<<<
021732 104460                                                                           TRAP    C$ERROR
021734 000415                           BR      60$             ;EXIT THE TEST.
```

```
5343
5344                                        ;REPORT THAT RX.DATA.VALID BIT WAS NOT SET AFTER A RESET COMPLETION.
5345  021736  012737  001276  002666  6$:     MOV     #0702.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5346  021744  012701  007052                  MOV     #EM0702,R1      ;SELECT ERROR MESSAGE.
5347  021750                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0702 <<<<<
5348  021750  104460                                                                                  TRAP    C$ERROR
5349  021752  000406                          BR      60$             ;EXIT THE TEST.
5350
5351                                        ;REPORT THAT RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO.
5352  021754  012737  001277  002666  8$:     MOV     #0703.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5353  021762  012701  007232                  MOV     #EM0703,R1      ;SELECT ERROR MESSAGE.
5354  021766                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0703 <<<<<
5355  021766  104460                                                                                  TRAP    C$ERROR
5356
5357  021770                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
5358  021770  012700  000340                                                                          MOV     #PRI07,R0
5359  021774  104441                                                                                  TRAP    C$SPRI
5360  021776  005037  002310                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5361  022002                                  ENDTST
5362  022002                                                                          L10031:
5363  022002  104401                                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RLNA -
                                ;++ ************************************************************************************
                                ;*              - RBUF RX LINE NUMBER FIELD TEST -
                                ;*      THIS TEST VERIFIES THAT THE DUT RBUF RX LINE NUMBER FIELD IS WORKING
                                ;*      CORRECTLY BY UTILIZING THE SELFTEST CODES WHICH ARE PUT IN THE RX
                                ;*      FIFO AFTER A BOARD RESET.
                                ;*
                                ;-- ************************************************************************************
                                        BGNTST
                                                                                T8::
                                        SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
022004  012700  000240                                                   MOV     #PRI05,R0
022010  104441                                                          TRAP    C$SPRI
        000010
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
022012  012737  000010  002274          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (8)
022020  012737  177777  002310          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
022026  012737  000001  002664          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
022034  012737  007415  002670          MOV     #EM0801,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
022042  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
022046  012702  000040                  MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
022052  005003                          CLR     R3              ;WAITING FOR BIT TO CLEAR.
022054  013704  002214                  MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
022060  010214                          MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
022062  004737  016100                  JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
022066  004737  014166                  JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
022072  103016                          BCC     4$              ;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.
                                ;-
022074  005001                          CLR     R1              ;CLEAR THE LINE COUNTER.
022076  012400                          MOV     (R4)+,R0        ;INCREMENT POINTER TO PNT TO THE DUT RBUF REG.
022100  011402                  2$:     MOV     (R4),R2         ;READ A CHARACTER FROM THE DUT RX FIFO.
022102  010203                          MOV     R2,R3
022104  000303                          SWAB    R3
022106  042703  177760                  BIC     #177760,R3      ;REMOVE ALL BUT LINE NUMBER BITS.
022112  020301                          CMP     R3,R1           ;COMPARE WITH EXPECTED LINE NUMBER.
022114  001017                          BNE     6$              ;GO REPORT ERROR IF LINE NUMBERS DON'T MATCH.
022116  005201                          INC     R1              ;INCREMENT THE EXPECTED LINE NUMBER.
022120  020127  000010                  CMP     R1,#NUMLNS      ;COMPARE WITH NUMBER OF LINES ON DUT.
022124  001365                          BNE     2$              ;LOOP UNTIL CODES FOR ALL LINES ARE READ.
022126  000423                          BR      60$             ;EXIT TEST WITHOUT ERROR.

                                ;+
                                ; ERROR REPORTS:
                                ;-
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
022130  012737  001441  002666  4$:      MOV     #0801.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
022136  012737  012576  002672          MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
022144  012701  004676                  MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
022150                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0801 <<<<<
```

```
5420  022150  104460                                                          TRAP    C$ERROR
5421  022152  000411                    BR      60$             ;EXIT THE TEST.
5422
5423                                ;REPORT THAT RX LINE NUMBER FIELD IS WRONG FOR SELFTEST CODE.
5424  022154  012737  001442  002666  6$:     MOV     #0802.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5425  022162  012737  012514  002672          MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5426  022170  012701  007455                  MOV     #EM0802,R1      ;SELECT ERROR MESSAGE.
5427  022174                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0802 <<<<<
5428  022174  104460                                                          TRAP    C$ERROR
5429
5430  022176                            60$:  SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
5431  022176  012700  000340                                                  MOV     #PRI07,R0
5432  022202  104441                                                          TRAP    C$SPRI
5433  022204  005037  002310                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5434  022210                                  ENDTST
5435  022210                                                        L10032:
5436  022210  104401                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - BMPCHK -
                                ;++ ************************************************************************
                                ;*              - BMP CHECK TEST -
                                ;*      THIS TEST IS USED TO VERIFY THAT THE DUT DOES NOT IMMEDIATELY FAIL
                                ;*      THE ON-BOARD BACKGROUND-MONITOR PROGRAM, AND HENCE INVALIDATE
                                ;*      SUCCEEDING TESTS.
                                ;*      THIS TEST LOOKS FOR BMP CODES IN THE FIFO FOR A SET PERIOD IMMEDIATELY
                                ;*      AFTER THE SELF-TEST IS SKIPPED.
                                ;*      ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE QUEUE AND ARE ALSO
                                ;*      REPORTED IN THIS TEST.
                                ;*
                                ;-- ************************************************************************
022212                                  BGNTST
022212                                                                          T9::
022212                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
022212  012700  000240                                                        MOV     #PRI05,R0
022216  104441                                                                TRAP    C$SPRI
        000011
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
022220  012737  000011  002274          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (9)
022226  012737  177777  002310          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
022234  012737  000001  002664          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
022242  012737  001605  002666          MOV     #0901.,ERRNBR   ;SET THE ERROR NUMBER.
                                ;+
                                ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                ;-
022250  012701  005670                  MOV     #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
022254  012702  000040                  MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
022260  005003                          CLR     R3              ;WAITING FOR BIT TO CLEAR.
022262  013704  002214                  MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
022266  004737  014166                  JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
022272  103027                          BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
                                ;+
                                ; RESET THE DUT, SKIP THE SELF-TEST.
                                ;-
022274  010214                          MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
022276  004737  016100                  JSR     PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
                                ;+
                                ; WAIT FOR MASTER RESET TO CLEAR. DELAY FOR 500 MILLI-SECS BEFORE PURGING
                                ; THE FIFO.
                                ;-
022302  012704  000764                  MOV     #500.,R4        ;TIME-OUT VALUE IS 500 MILLI-SECONDS.
022306  004737  014126                  JSR     PC,DELAY        ;WAIT FOR BMP TO BEGIN EXECUTION.
022312  004737  014544                  JSR     PC,PUFIFO       ;PURGE THE FIFO, SAVING ANY BMP CODES.
022316  103015                          BCC     50$             ;ABORT THE TEST IF THE FIFO DID NOT CLEAR.
                                ;+
                                ; REPORT THE ERROR IF ANY BMP CODES WERE FOUND.
                                ;-
022320  013702  002462                  MOV     BMPCQP,R2       ;GET THE CONTENTS OF THE POINTER TO THE BMP Q.
022324  012703  002464                  MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
022330  020203                          CMP     R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
022332  001414                          BEQ     60$             ;EXIT NO CODES IN THE QUEUE.
                                ;+
                                ; THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
                                ;-
                                        ;REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN''
```

```
 5493  022334  012701  007561                    MOV     #EM0902,R1      ;PASS THE MESSAGE TO BE REPORTED.
 5494  022340                                    ERRDF   0901,EM0901,ER9301 ;            >>>>> ERROR #0901 <<<<<.
 5495  022340  104455                                                                    TRAP    C$ERDF
 5496  022342  001605                                                                    .WORD   901
 5497  022344  007530                                                                    .WORD   EM0901
 5498  022346  013372                                                                    .WORD   ER9301
 5499  022350  000405                            BR      60$
 5500
 5501  022352  012737  001606  002666  50$:      MOV     #902.,ERRNBR    ;SET >>>>> ERROR #0902 <<<<<.
 5502  022360  004737  016226                    JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
 5503
 5504  022364                            60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
 5505  022364  012700  000340                                                           MOV     #PRI07,R0
 5506  022370  104441                                                                   TRAP    C$SPRI
 5507  022372  005037  002310                    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5508  022376                                    ENDTST
 5509  022376                                                                  L10033:
 5510  022376  104401                                                                   TRAP    C$ETST
```

```
5511
5512
5513                                    .SBTTL  HARDWARE TEST           - SKSELF -
5514                                    ;++ ************************************************************************
5515                                    ;*              - SKIP SELF-TEST TEST -
5516                                    ;*      THIS TEST VERIFIES THAT THE DUT SKIPS THE SELF-TEST WITHIN THE
5517                                    ;*      TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS
5518                                    ;*      COMPLETION.
5519                                    ;*
5520                                    ;-- ************************************************************************
5521  022400                                    BGNTST
5522  022400                                                                          T10::
5523  022400                                    SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
5524  022400  012700  000240                                                        MOV     #PRI05,R0
5525  022404  104441                                                                TRAP    C$SPRI
5526          000012                     TNUM == TNUM + 1              ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5527  022406  012737  000012  002274           MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (10)
5528  022414  012737  177777  002310           MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
5529  022422  012737  000001  002664           MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5530  022430  012737  007615  002670           MOV     #EM1001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5531  022436  012737  012576  002672           MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5532                                    ;+
5533                                    ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5534                                    ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5535                                    ;-
5536  022444  012701  005670                   MOV     #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
5537  022450  012702  000040                   MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
5538  022454  005003                           CLR     R3              ;WAITING FOR BIT TO CLEAR.
5539  022456  013704  002214                   MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
5540  022462  004737  014166                   JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5541  022466  103037                           BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
5542                                    ;+
5543                                    ; DETERMINE IF THE DUT TAKES TOO SHORT OR TOO LONG A TIME TO SKIP THE SELF-TEST
5544                                    ;  SET-UP A TIME-OUT OF 50 MILLI-SECOND, IF MR IS CLEAR IN LESS THAN 10 MILLI
5545                                    ;   -SECOND, OR GREATER THAN 50 MILLI-SECONDS, REPORT THE ERROR.
5546                                    ;-
5547  022470  012701  000062                   MOV     #50.,R1         ;TIME-OUT VALUE IS 50 MILLI-SECONDS.
5548  022474  010214                           MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
5549  022476  004737  016100                   JSR     PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
5550  022502  004737  014166                   JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5551  022506  103011                           BCC     2$              ;GO REPORT ERR IF SKIPPING STEST TOOK TOO LONG.
5552  022510  020127  000050                    CMP     R1,#40.
5553  022514  003015                           BGT     4$              ;GO REP ERR IF SELFTEST COMPLETED IN < 10 MS.
5554                                    ;+
5555                                    ; SELF-TEST COMPLETED WITHIN 10 MILLI-SEC TO 50 MILLI-SECONDS.
5556                                    ;  VERIFY THAT THE SELF-TEST CODES IN THE FIFO ARE "GOOD" CODES ,IE THE DUT
5557                                    ;  SUCCESSFULLY COMPLETED THE SELF-TEST.
5558                                    ; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 1003 THRU 1007 <<<<<.
5559                                    ;-
5560  022516  012737  001753  002666           MOV     #1003.,ERRNBR   ;SET ERROR NUMBER TO 1003.
5561  022524  004737  015450                   JSR     PC,RSTRPT       ;CHECK SELF-TEST CODES IN THE FIFO.
5562  022530  000423                           BR      60$             ;EXIT TEST.
5563                                    ;+
5564                                    ; ERROR REPORTS:
5565                                    ;-
5566                                        ;REPORT SKIP SELF-TEST TOOK TOO LONG.
```

```
5567  022532  012737  001751  002666  2$:     MOV     #1001.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5568  022540  012701  007661                  MOV     #EM1002,R1      ;SELECT ERROR MESSAGE.
5569  022544                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #1001 <<<<<
5570  022544  104460                                                                                  TRAP    C$ERROR
5571  022546  000414                          BR      60$             ;EXIT THE TEST.
5572
5573                                  ;REPORT SKIP SELF-TEST COMPLETED TOO SOON.
5574  022550  012737  001752  002666  4$:     MOV     #1002.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5575  022556  012701  007746                  MOV     #EM1003,R1      ;SELECT ERROR MESSAGE.
5576  022562                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #1002 <<<<<
5577  022562  104460                                                                                  TRAP    C$ERROR
5578  022564  000405                          BR      60$             ;EXIT THE TEST.
5579
5580  022566  012737  001753  002666  50$:    MOV     #1003.,ERRNBR   ;SET ERROR NUMBER.
5581  022574  004737  016226                  JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
5582
5583  022600                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
5584  022600  012700  000340                                                          MOV     #PRI07,R0
5585  022604  104441                                                                 TRAP    C$SPRI
5586  022606  005037  002310                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5587  022612                                  ENDTST
5588  022612                                                                  L10034:
5589  022612  104401                                                                  TRAP    C$ETST
```

```
5590
5591
5592                                    .SBTTL  HARDWARE TEST           - DFSKST -
5593                                    ;++ ************************************************************************
5594                                    ;*              - DIAGNOSTIC FAIL BIT, SKIP SELF-TEST TEST -
5595                                    ;*      THIS TEST VERIFIES THAT THE DIAGNOSTIC FAIL BIT OF THE DUT, CORRECTLY
5596                                    ;*      CHANGES STATE AS THE ON-BOARED SELFTEST IS SKIPPED.
5597                                    ;*
5598                                    ;-- ************************************************************************
5599  022614                                    BGNTST
5600  022614                                                                                    T11::
5601  022614                                    SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
5602  022614  012700  000240                                                                   MOV     #PRI05,R0
5603  022620  104441                                                                           TRAP    C$SPRI
5604          000013                            TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5605  022622  012737  000013  002274            MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (11)
5606  022630  012737  177777  002310            MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
5607  022636  012737  000001  002664            MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5608  022644  012737  010024  002670            MOV     #EM1101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5609  022652  012737  012576  002672            MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5610                                    ;+
5611                                    ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5612                                    ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5613                                    ;-
5614  022660  012701  005670                    MOV     #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
5615  022664  012702  000040                    MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
5616  022670  005003                            CLR     R3              ;WAITING FOR BIT TO CLEAR.
5617  022672  013704  002214                    MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
5618  022676  004737  014166                   JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5619  022702  103044                            BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
5620                                    ;+
5621                                    ; RESET THE DUT, SKIP THE SELF-TEST.
5622                                    ;-
5623  022704  010214                            MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
5624  022706  004737  016100                   JSR     PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
5625                                    ;+
5626                                    ; SET TIME OUT OF 5 MILLI SECONDS, WAIT FOR DIAG_FAIL BIT TO SET.
5627                                    ;  IF TIME-OUT OCCURS GO REPORT THE ERROR.
5628                                    ;-
5629  022712  012701  000005                    MOV     #5,R1           ;TIME-OUT VALUE IS 5 MILLI-SECONDS.
5630  022716  012702  020000                    MOV     #BIT13,R2       ;WAITING FOR DIAGNOSTIC FAIL BIT.
5631  022722  010203                            MOV     R2,R3           ;WAITING FOR BIT TO SET.
5632  022724  013704  002214                    MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
5633  022730  004737  014166                   JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
5634  022734  103020                            BCC     4$              ;IF DIAG_FAIL DID NOT SET, GO REPORT ERROR.
5635
5636                                    ;+
5637                                    ; SET TIME-OUT OF 15 MILLI-SECS, WAIT FOR DIAG_FAIL TO CLEAR.
5638                                    ;  IF TIME-OUT OCCURS GO REPORT THE ERROR.
5639                                    ;  VERIFY THE DIAG_FAIL BIT IS IN A STABLE STATE BEFORE CONTINUING.  LOOP
5640                                    ;  BACK IF THE STATE WAS TRANSITORY, USING THE REMAINDER OF THE 15 MS TIME-OUT.
5641                                    ;-
5642  022736  012701  000017                    MOV     #15.,R1         ;TIME-OUT VALUE IS 15 MILLI-SECONDS.
5643  022742  005003                   2$:      CLR     R3              ;WAITING FOR BIT TO CLEAR.
5644  022744  004737  014166                   JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
5645  022750  103012                            BCC     4$              ;IF DIAG_FAIL DID NOT CLEAR, GO REPORT ERROR.
```

```
5646  022752  010105                          MOV     R1,R5           ;SAVE THE REMAINING TIME-OUT VALUE.
5647  022754  012701  000001                  MOV     #1,R1           ;SET TIME-OUT OF 1 MILLI-SECOND.
5648  022760  052703  020000                  BIS     #BIT13,R3       ;WAIT FOR BIT TO SET.
5649  022764  004737  014166                  JSR     PC,MSLGET       ;DOUBLE CHECK TO ELIMINATE NOISE PROBLEMS.
5650  022770  103016                          BCC     60$             ;EXIT IF DIAG FAIL BIT STILL CLEAR.
5651  022772  010501                          MOV     R5,R1           ;PASS THE REMAINING TIME-OUT VALUE.
5652  022774  000762                          BR      2$              ;LOOP TO CHECK AGAIN.
5653                                  ;+
5654                                  ; ERROR REPORTS:
5655                                  ;-
5656                                      ;REPORT DIAGNOSTIC FAI' BIT BAD.
5657  022776  012737  002115  002666  4$:   MOV     #1101.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5658  023004  012701  010265                  MOV     #EM1205,R1      ;SELECT ERROR MESSAGE.
5659  023010                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #1101 <<<<<
5660  023010  104460                                                                                  TRAP    C$ERROR
5661  023012  000405                          BR      60$             ;EXIT THE TEST.
5662
5663  023014  012737  002116  002666  50$:    MOV     #1102.,ERRNBR   ;SET THE ERROR NUMBER FOR TSABRT RTN.
5664  023022  004737  016226                  JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
5665
5666  023026                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
5667  023026  012700  000340                                                                          MOV     #PRI07,R0
5668  023032  104441                                                                                  TRAP    C$SPRI
5669  023034  005037  002310                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED A TEST.
5670  023040                                  ENDTST
5671  023040                                                                                  L10035:
5672  023040  104401                                                                                  TRAP    C$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.
                                ;*
                                ;-- ************************************************************************
023042                          BGNTST
023042                                                                  T12::
023042                          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
023042  012700  000240                                                   MOV     #PRI05,R0
023046  104441                                                          TRAP    C$SPRI
        000014                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
023050  012737  000014  002274   MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (12)
023056  012737  177777  002310   MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
023064  012737  000001  002664   MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
023072  012737  010050  002670   MOV    #EM1201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
023100  012737  012576  002672   MOV    #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                ;-
023106  012701  005670           MOV    #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
023112  012702  000040           MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
023116  005003                   CLR    R3              ;WAITING FOR BIT TO CLEAR.
023120  013704  002214           MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
023124  004737  014166          JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
023130  103062                   BCC    50$             ;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.
                                ;-
023132  012701  005670           MOV    #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
023136  010214                   MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
023140  004737  014166          JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
023144  103030                  BCC     4$              ;GO REPORT ERROR SELFTEST TOOK TOO LONG.
023146  012702  005670           MOV    #3000.,R2
023152  160102                   SUB    R1,R2           ;CALCULATE # OF MS SELFTEST TO COMPLETE.
023154  020227  000062           CMP    R2,#50.
023160  002431                  BLT     6$              ;SELFTEST SKIPPED?  YES, GO REPORT ERROR.
023162  020227  000764           CMP    R2,#500.
023166  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.
                                ;-
023170  032714  020000           BIT    #BIT13,(R4)     ;DETERMINE IF THE DIAG FAIL BIT IS CLEAR.
023174  001406                  BEQ     2$              ;SKIP ERROR REPORT IF BIT IS CLEAR.
                                ;REPORT DIAGNOSTIC FAIL BIT BAD.
023176  012737  002264  002666   MOV    #1204.,ERRNBR   ;SET ERROR NUMBER TO IN ERROR TABLE.
023204  012701  010265           MOV    #EM1205,R1      ;SELECT THE ERROR MESSAGE.
023210                          ERROR                   ;               >>>>> ERROR #1204 <<<<<
023210  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 <<<<<.
                                        ;-
023212  012737  002265  002666  2$:     MOV     #1205.,ERRNBR   ;SET ERROR NUMBER TO 1205.
023220  004737  015450                  JSR     PC,RSTRPT       ;CHECK SELF-TEST CODES IN THE FIFO.
023224  000431                          BR      60$             ;EXIT TEST.
                                        ;+
                                        ; ERROR REPORTS:
                                        ;-
                                            ;REPORT SELF-TEST TOOK TOO LONG TO COMPLETE.
023226  012737  002261  002666  4$:       MOV     #1201.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
023234  012701  010067                    MOV     #EM1202,R1      ;SELECT ERROR MESSAGE.
023240                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #1201 <<<<<
023240  104460                                                                              TRAP    C$ERROR
023242  000422                          BR      60$             ;EXIT THE TEST.

                                            ;REPORT SELF-TEST DID NOT EXECUTE AFTER DUT RESET.
023244  012737  002262  002666  6$:       MOV     #1202.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
023252  012701  010231                    MOV     #EM1204,R1      ;SELECT ERROR MESSAGE.
023256                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #1202 <<<<<
023256  104460                                                                              TRAP    C$ERROR

                                            ;REPORT SELF-TEST COMPETED TOO SOON.
023260  012737  002263  002666  8$:       MOV     #1203.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
023266  012701  010153                    MOV     #EM1203,R1      ;SELECT ERROR MESSAGE.
023272                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #1203 <<<<<
023272  104460                                                                              TRAP    C$ERROR
023274  000405                          BR      60$             ;EXIT THE TEST.

023276  012737  002272  002666  50$:     MOV     #1210.,ERRNBR   ;SET THE ERROR NUMBER FOR TSABRT RTN.
023304  004737  016226                  JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.

023310                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
023310  012700  000340                                                          MOV     #PRI07,R0
023314  104441                                                                  TRAP    C$SPRI
023316  005037  002310                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
023322                                  ENDTST
023322                                                                  L10036:
023322  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.
                                        ;*
                                        ;-- ************************************************************************
023324                                          BGNTST
023324                                                                          T13::
023324                                          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
023324  012700  000240                                                          MOV     #PRI05,R0
023330  104441                                                                  TRAP    C$SPRI
        000015                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
023332  012737  000015  002274           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (13)
023340  012737  177777  002310           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
023346  012737  000001  002664           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
023354  012737  010311  002670           MOV    #EM1301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
023362  012737  012576  002672           MOV    #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
023370  012737  002425  002666           MOV    #1301.,ERRNBR   ;SET ERROR NUMBER TO 1301.
                                        ;+
                                        ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                        ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                        ;-
023376  012701  005670                           MOV    #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
023402  012702  000040                           MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
023406  005003                                   CLR    R3              ;WAITING FOR BIT TO CLEAR.
023410  013704  002214                           MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
023414  004737  014166                          JSR    PC,MSLGET       ;WAIT FOR DUT CSR_MR BIT TO CLEAR.
023420  103071                                   BCC    50$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                        ;+
                                        ; RESET THE DUT, DELAY FOR 25 MILLI-SECONDS BEFORE WRITING THE FAIL_SELF_TEST
                                        ; CODE TO TBUFFCT REGISTER ON CHANNEL 0.
                                        ;-
023422  010214                                   MOV    R2,(R4)         ;SET DUT MASTER RESET BIT, SELECT CHANNEL 0.
023424  012704  000031                           MOV    #25.,R4         ;PASS DELAY PERIOD OF 25 MILLI SECS.
023430  004737  014126                          JSR    PC,DELAY        ;WAIT FOR SELFTEST TO INITIALISE.
023434  012777  146314  156570                   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.
                                        ;-
023442  005237  002666                           INC    ERRNBR          ;SET ERROR NUMBER TO 1302.
023446  012701  003720                           MOV    #2000.,R1       ;TIME-OUT VALUE IS 2.0 SECONDS.
023452  013704  002214                           MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
023456  004737  014166                          JSR    PC,MSLGET       ;WAIT FOR DUT CSR_MR BIT TO CLEAR.
023462  103050                                   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.
                                        ;-
023464  005237  002666                           INC    ERRNBR          ;SET ERROR NUMBER TO 1303.
023470  032714  020000                           BIT    #BIT13,(R4)     ;CHECK THE STATE OF THE DIAG_FAIL BIT.
023474  001437                                  BEQ    8$              ;GO REPORT ERROR IF DIAG_FAIL BIT CLEAR.
```

```
5825                                    ;+
5826                                    ; REMOVE THE 8 SELF-TEST CODES FORM THE FIFO, AND VERIFY THAT AT LEAST
5827                                    ; ONE IS A PROC1 TO RAM ERROR CODE (231).
5828                                    ;-
5829  023476  005237  002666                   INC     ERRNBR          ;SET ERROR NUMBER TO 1304.
5830  023502  012700  000010                   MOV     #8.,R0          ;SET MAXIMUM READ COUNT.
5831  023506  005001                           CLR     R1              ;CLEAR THE CORRECT CODE COUNTER.
5832  023510  005003                           CLR     R3              ;CLEAR THE ROM VERSION 0 FLAG.
5833  023512  013704  002216                   MOV     RBUFA,R4        ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
5834  023516  011402                  2$:      MOV     (R4),R2         ;READ A CODE FROM THE FIFO.
5835  023520  100031                          BPL     50$             ;GO REPORT ERROR IF THE FIFO IS EMPTY.
5836  023522  042702  007400                   BIC     #7400,R2        ;REMOVE THE LINE NUMBER FROM THE CODE.
5837  023526  120227  170231                   CMPB    R2,#170231      ;IS IT THE CORRECT ERROR CODE?.
5838  023532  001001                          BNE     4$              ;SKIP NEXT INSTRUCTION, IF NOT A 231 CODE.
5839  023534  005201                           INC     R1              ;INCREMENT COUNTER.
5840  023536  042702  000002          4$:      BIC     #BIT1,R2        ;REMOVE PROCESSOR INDICATOR BIT.
5841  023542  020227  170001                   CMP     R2,#170001      ;COMPARE WITH ROM VERSION #0 CODE.
5842  023546  001002                          BNE     6$              ;ROM VERSION #0?  NO, SKIP SETTING FLAG.
5843  023550  012703  177777                   MOV     #-1,R3          ;YES, SET THE ROM VERSION #0 FLAGS.
5844  023554  005300                  6$:      DEC     R0              ;DECREMENT MAX READ COUNTER.
5845  023556  001357                          BNE     2$              ;LOOP IF 8 CODES HAVE NOT BEEN READ.
5846  023560  005701                           TST     R1              ;WERE ANY 231 CODES FOUND?.
5847  023562  001012                          BNE     60$             ;YES, THEN EXIT.
5848  023564  005703                           TST     R3              ;CHECK THE ROM VERSION #1 INDICATOR.
5849  023566  001010                          BNE     60$             ;ROM VERSION 0 IN EITHER PROCESSOR?  YES, EXIT.
5850  023570  005237  002666                   INC     ERRNBR          ;NO, SET ERROR NUMBER TO 1305 AND REPORT ERROR.
5851                                  ;REPORT SELF-TEST ERROR REPORTING BAD.
5852  023574  012701  010335          8$:      MOV     #EM1302,R1      ;SELECT ERROR MESSAGE.
5853  023600                                   ERROR                   ;REPORT ERROR.          >>>>> ERROR <<<<<
5854  023600  104460                                                                                   TRAP    C$ERROR
5855  023602  000402                          BR      60$             ;EXIT THE TEST.
5856
5857  023604  004737  016226          50$:     JSR     PC,TSABRT       ;REPORT NON-RELATED TEST ERROR.
5858
5859  023610                          60$:     SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
5860  023610  012700  000340                                                                   MOV     #PRI07,R0
5861  023614  104441                                                                           TRAP    C$SPRI
5862  023616  005037  002310                   CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5863  023622                                   ENDTST
5864  023622                                                                           L10037:
5865  023622  104401                                                                           TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE TEST           - ROMVER -
                                        ;++ ************************************************************************
                                        ;*              - ROM VERSION TEST -
                                        ;*      THIS TEST VERIFIES THAT THE DUT'S SELF-TEST PLACES VALID ROM VERSION
                                        ;*      NUMBERS IN THE FIFO AFTER IT HAS BEEN SKIPPED.  THE ROM VERSION NUMBERS
                                        ;*      WILL BE REPORTED (ON THE FIRST PASS ONLY), IF AN AFFIRMATIVE ANSWER
                                        ;*      WAS GIVEN TO THE SOFTWARE P-TABLE QUESTION.
                                        ;*
                                        ;-- ************************************************************************
023624                                          BGNTST
                                                                                T14::
023624                                          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
023624  012700  000240                                                    MOV     #PRI05,R0
023630  104441                                                          TRAP    C$SPRI
        000016                           TNUM == TNUM + 1               ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
023632  012737  000016  002274          MOV     #TNUM,TSTNUM           ;SET UP THE TEST NUMBER.        (14)
023640  012737  177777  002310          MOV     #-1,CTRLCF             ;INDICATE THAT WE ARE WITHIN A TEST.
023646  012737  000001  002664          MOV     #1,ERRTYP              ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
023654  012737  010374  002670          MOV     #EM1401,ERRMSG         ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
023662  012737  012576  002672          MOV     #ER0503,ERRBLK         ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                        ;+
                                        ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                        ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                        ;-
023670  012701  005670                          MOV     #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
023674  012702  000040                          MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
023700  005003                                  CLR     R3              ;WAITING FOR BIT TO CLEAR.
023702  013704  002214                          MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
023706  004737  014166                        JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
023712  103131                                  BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
                                        ;+
                                        ; SET THE MASTER RESET BIT, AND SKIP THE SELF TEST.
                                        ;-
023714  010214                                MOV     R2,(R4)         ;SET THE MASTER RESET BIT.
023716  004737  016100                        JSR     PC,SKPSTS       ;SKIP THE SELF TEST.
023722  012701  005670                          MOV     #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
023726  004737  014166                        JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
023732  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.
                                        ;-
023734  012705  000040                          MOV     #4*NUMLNS,R5    ;SET MAXIMUM TRY COUNTER.
023740  012703  000143                          MOV     #99.,R3         ;SET AN INVALID ROM VERSION NUMBER FOR PROC_1.
023744  010304                                  MOV     R3,R4           ;SET AN INVALID ROM VERSION NUMBER FOR PROC_2.
023746  012737  002571  002666                  MOV     #1401.,ERRNBR   ;SET THE ERROR NUMBER TO 1401.
023754  012701  010424                          MOV     #EM1402,R1      ;SELECT MESSAGE TO BE REPORTED IF FIFO EMPTY.

023760  017702  156232          2$:     MOV     @RBUFA,R2       ;READ THE NEXT CHAR FROM THE FIFO.
023764  100077                        BPL     12$             ;GO REPORT ERROR IF FIFO EMPTY.
                                          ;+
```

```
5922                                    ; CHECK IF THE READ DATA IS A BMP CODE.
5923                                    ;-
5924  023766  012700  000301                MOV     #301,R0         ;SET-UP A BIT MASK OF A BMP CODE.
5925  023772  040200                        BIC     R2,R0           ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
5926  023774  001003                      BNE     4$                ;BRANCH IF NOT A BMP CODE.
5927  023776  004737  016032              JSR     PC,SAVBMP         ;SAVE THE BMP CODE ON THE QUEUE.
5928  024002  000435                      BR      8$                ;
5929                                    ;+
5930                                    ; CHECK IF THE READ DATA IS A SELF-TEST CODE.
5931                                    ;-
5932  024004  012700  000201          4$:    MOV     #201,R0         ;SET-UP A BIT MASK OF A SELFTEST CODE.
5933  024010  040200                        BIC     R2,R0           ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
5934  024012  001431                      BEQ     8$                ;BRANCH IF IT IS A SELFTEST CODE.
5935
5936                                    ;+
5937                                    ; THE READ DATA IS A ROM VERSION NUMBER, DETERMINE WHICH ONE IT IS.
5938                                    ;-
5939  024014  032702  000002                BIT     #BIT1,R2        ;CHECK THE PROCESSOR NUMBER BIT IN THE CODE.
5940  024020  001407                      BEQ     6$                ;BRANCH IF IT IS PROC_1 ROM VERSION NUMBER.
5941  024022  010204                        MOV     R2,R4           ;SAVE PROC_2 ROM VERSION NUMBER.
5942  024024  042704  177603                BIC     #177603,R4      ;CLEAR ANY UNWANTED BITS.
5943  024030  000241                        CLC                     ;CLEAR THE CARRY BIT.
5944  024032  006004                        ROR     R4              ;SHIFT THE CODES ALONG TO GET THE ROM
5945  024034  006004                        ROR     R4              ; VERSION NUMBER IN THE LOW 5 BITS.
5946  024036  000417                      BR      8$                ;
5947  024040  010203                   6$:    MOV     R2,R3           ;SAVE PROC_1 ROM VERSION NUMBER.
5948  024042  042703  177603                BIC     #177603,R3      ;CLEAR ANY UNWANTED BITS.
5949  024046  000241                        CLC                     ;CLEAR THE CARRY BIT.
5950  024050  006003                        ROR     R3              ;SHIFT THE CODE ALONG TO GET THE ROM
5951  024052  006003                        ROR     R3              ; VERSION NUMBER IN THE LOW 5 BITS.
5952  024054  020427  000143                CMP     R4,#99.         ;CHECK IF WE HAVE RECEIVE PROC_2 ROM CODE.
5953  024060  001016                      BNE     10$               ;GO REPORT BOTH ROM VERSION NUMBERS.
5954                                  ;+
5955                                  ; RECEIVED ROM VERSION NUMBERS OUT OF SEQUENCE.
5956                                  ;  IE, PROC_1'S ROM VERSION NUMBER FOUND IN THE FIFO BEFORE PROC_2'S.
5957                                  ;-
5958  024062  012701  010512                MOV     #EM1403,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5959  024066  012737  002572  002666        MOV     #1402.,ERRNBR   ;SET THE ERROR NUMBER.
5960  024074  000433                      BR      12$               ;GO REPORT ERROR.
5961
5962  024076  005305                   8$:    DEC     R5              ;DECREMENT THE MAX TRY COUNTER.
5963  024100  001327                      BNE     2$                ;LOOP TO GET THE NEXT CHAR FROM THE FIFO.
5964  024102  012701  010565                MOV     #EM1404,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5965  024106  012737  002573  002666        MOV     #1403.,ERRNBR   ;SET THE ERROR NUMBER.
5966  024114  000423                        BR      12$             ;GIVE UP, GO REPORT ERROR.
5967                                  ;+
5968                                  ; IF THIS IS THE FIRST PASS, AND SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
5969                                  ; THEN REPORT THE ROM VERSION NUMBERS TO THE OPERATOR.
5970                                  ;-
5971  024116  032737  000001  002204  10$:    BIT     #BIT0,OPTION    ;CHECK ON THE STATE OF THE SOFTWARE SWITCH.
5972  024124  001431                      BEQ     60$               ;EXIT IF NO ROM VERSION PRINTOUT WAS REQUESTED.
5973  024126  023727  002300  000001        CMP     PASCNT,#1       ;CHECK IF THIS IS THE FIRST PASS.
5974  024134  003025                      BGT     60$               ;EXIT IF ROM VERS HAVE ALREADY BEEN REPORTED.
5975  024136                               PRINTB #EF1401,R3,R4    ;PRINT THE ROM VERSION NUMBERS.
5976  024136  010446                                                                   MOV     R4,-(SP)
5977  024140  010346                                                                   MOV     R3,-(SP)
```

```
 5978  024142  012746  003035                                                          MOV     #EF1401,-(SP)
 5979  024146  012746  000003                                                          MOV     #3,-(SP)
 5980  024152  010600                                                                 MOV     SP,R0
 5981  024154  104414                                                                 TRAP    C$PNTB
 5982  024156  062706  000010                                                         ADD     #10,SP
 5983  024162  000412                      BR      60$             ;EXIT THIS TEST.
 5984                                ;-
 5985                                ; ERROR REPORTS:
 5986                                ;-
 5987  024164  012737  012622  002672  12$:   MOV     #ER1401,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
 5988  024172                              ERROR                   ;REPORT ERROR.          >>>>> ERROR <<<<<
 5989  024172  104460                                                                 TRAP    C$ERROR
 5990  024174  000405                      BR      60$
 5991
 5992  024176  012737  002575  002666  50$:   MOV     #1405.,ERRNBR   ;SET UP ERROR NUMBER FOR TSABRT RTN.
 5993  024204  004737  016226              JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
 5994
 5995  024210                      60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
 5996  024210  012700  000340                                                         MOV     #PRI07,R0
 5997  024214  104441                                                                 TRAP    C$SPRI
 5998  024216  005037  002310              CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5999  024222                              ENDTST
 6000  024222                                                                  L10040:
 6001  024222  104401                                                                 TRAP    C$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 #PRI05          ;ALLOW THE LTC TO INTERRUPT.
024224                                                                          MOV     #PRI05,R0
024224                                                                          TRAP    C$SPRI
024224  012700  000240
024230  104441
                000017
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
024232  012737  000017  002274          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (16)
024240  012737  177777  002310          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
024246  012737  000001  002664          MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
024254  012737  003101  002666          MOV     #1601.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
024262  012737  010770  002670          MOV     #EM1604,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
024270  005037  002420                  CLR     ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
024274  012700  002422                  MOV     #ERCNTB,R0
024300  004737  014026                 JSR     PC,CLR16W       ;CLEAR THE ERROR COUNTER TABLE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1601 <<<<<.
                                ;-
024304  004737  015304                  JSR     PC,RESETT       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
024310  103402                         BCS     .+6             ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
024312  000137  024434                 JMP     60$             ;YES,  EXIT THE TEST.
                                ;+
                                ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR
                                ;-
024316  005237  002666                  INC     ERRNBR          ;SET THE ERROR REPORT NUMBER TO 1602.
024322  012702  000017                  MOV     #17,R2          ;SET LOOP COUNT.
024326  013704  002214                  MOV     CSRA,R4         ;GET CSR ADDRESS.
024332  010214                 2$:      MOV     R2,(R4)         ;WRITE COUNT TO CSR.
024334  011401                          MOV     (R4),R1         ;READ BACK THE CONTENTS OF THE CSR
024336  042701  177760                  BIC     #177760,R1      ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
024342  020102                          CMP     R1,R2           ;CHECK FOR CORRECT DATA WRITTEN/READ.
024344  001406                         BEQ     4$              ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                    ;REPORT 'BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D).'
024346  012737  012752  002672          MOV     #ER1601,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
024354  005003                          CLR     R3              ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
024356  005005                          CLR     R5              ;CAUSE REPORT OF LINE 0.
024360                                  ERROR                   ;  >>>>> ERROR # 1602 <<<<<
024360  104460                                                                  TRAP    C$ERROR
024362  005302                 4$:      DEC     R2              ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
024364  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 <<<<<.
                                ;-
024366  005237  002666                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 1603.
```

```
6058  024372  005003                          CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6059  024374  012704  000002                  MOV     #2,R4           ;INDICATE R/W ACCESS, CLEAR FIRST.
6060  024400  004737  015046                  JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6061                                    ;+
6062                                    ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
6063                                    ;   ACTIVE LINES.  BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
6064                                    ;   REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1606 - 1608 <<<<<.
6065                                    ;-
6066  024404  012737  003106  002666            MOV     #1606.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6067  024412  005003                          CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6068  024414  005404                          NEG     R4              ;INDICATE R/W ACCESS, SET FIRST.
6069  024416  004737  015046                  JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6070                                    ;+
6071                                    ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6072                                    ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1609 <<<<<
6073                                    ;-
6074  024422  012737  003111  002666            MOV     #1609.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
6075  024430  004737  015256                  JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
6076  024434  005037  002310          60$:     CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
6077  024440                                  ENDTST
6078  024440                                                                  L10041:
6079  024440  104401                                                                  TRAP    C$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.
                                ;*
                                ;-- ************************************************************************

024442                                  BGNTST
                                                                                T16::
024442                           SETPRI #PRI05          ;ALLOW THE LTC TO INTERRUPT.
024442 012700  000240                                                   MOV     #PRI05,R0
024446 104441                                                           TRAP    C$SPRI
       000020                    TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
024450 012737  000020  002274    MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (17)
024456 012737  177777  002310    MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
024464 012737  000001  002664    MOV     #1,ERRTYP      ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
024472 012737  003245  002666    MOV     #1701.,ERRNBR  ;SET UP ERROR NUMBER IN THE ERROR TABLE.
024500 012737  011036  002670    MOV     #EM1701,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
024506 005037  002420            CLR     ERSMRF         ;CLEAR THE ERROR SUMMARY FLAGS.
024512 012700  002422            MOV     #ERCNTB,R0
024516 004737  014026            JSR     PC,CLR16W      ;CLEAR THE ERROR COUNTER TABLE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1701 <<<<<.
                                ;-
024522  004737  015304                  JSR     PC,RESETT      ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
024526  103402                          BCS     .+6            ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
024530  000137  024656                  JMP     60$            ;YES,  EXIT THE TEST.
                                ;+
                                ; VERIFY READ/MODIFY/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR
                                ;-
024534  005237  002666                   INC     ERRNBR        ;SET THE ERROR REPORT NUMBER TO 1702.
024540  012702  000017                   MOV     #17,R2        ;SET LOOP COUNT.
024544  013704  002214                   MOV     CSRA,R4       ;GET CSR ADDRESS.
024550  042714  000017           2$:     BIC     #17,(R4)      ;CLEAR THE DUT CSR USING READ/MODIFY/WRITE.
024554  050214                          BIS     R2,(R4)       ;WRITE COUNT TO CSR USING READ/MODIFY/WRITE.
024556  011401                          MOV     (R4),R1       ;READ BACK THE CONTENTS OF THE CSR
024560  042701  177760                  BIC     #177760,R1    ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
024564  020102                          CMP     R1,R2         ;CHECK FOR CORRECT DATA WRITTEN/READ.
024566  001406                          BEQ     4$            ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                    ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
024570  012737  012752  002672          MOV     #ER1601,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
024576  005003                          CLR     R3            ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
024600  005005                          CLR     R5            ;CAUSE REPORT OF LINE 0.
024602                                  ERROR                 ;  >>>>> ERROR # 1702 <<<<<
024602  104460                                                          TRAP    C$ERROR
024604  005302                   4$:     DEC     R2            ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
024606  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 <<<<<.
                                ;-
```

```
 6136  024610  005237  002666          INC     ERRNBR          ;SET THE ERROR NUMBER TO 1703.
 6137  024614  005003                  CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
 6138  024616  012704  000001          MOV     #1,R4           ;INDICATE R/M/W ACCESS, CLEAR FIRST.
 6139  024622  004737  015046          JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6140                                ;+
 6141                                ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
 6142                                ;   ACTIVE LINES USING R/M/W.  BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
 6143                                ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1706 - 1708 <<<<<.
 6144                                ;-
 6145  024626  012737  003252  002666          MOV     #1706.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
 6146  024634  005003                  CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
 6147  024636  005404                  NEG     R4              ;INDICATE R/M/W ACCESS, SET FIRST.
 6148  024640  004737  015046          JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6149                                ;+
 6150                                ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
 6151                                ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1709 <<<<<
 6152                                ;-
 6153  024644  012737  003255  002666          MOV     #1709.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
 6154  024652  004737  015256          JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
 6155  024656  005037  002310  60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 6156  024662                          ENDTST
 6157  024662                                                          L10042:
 6158  024662  104401                                                          TRAP    C$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 #PRI05          ;ALLOW THE LTC TO INTERRUPT.
024664  012700  000240                                                          MOV     #PRI05,R0
024670  104441                                                                  TRAP    C$SPRI
        000021                  TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
024672  012737  000021  002274  MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (18)
024700  012737  177777  002310  MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
024706  012737  000001  002664  MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
024714  012737  003411  002666  MOV     #1801.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
024722  012737  011113  002670  MOV     #EM1801,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
024730  005037  002420          CLR     ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
024734  012700  002422          MOV     #ERCNTB,R0
024740  004737  014026          JSR     PC,CLR16W       ;CLEAR THE ERROR COUNTER TABLE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1801 <<<<<.
                                ;-
024744  004737  015304          JSR     PC,RESETT       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
024750  103402                  BCS     .+6             ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
024752  000137  025130          JMP     60$             ;YES,  EXIT THE TEST.
024756  012737  003412  002666  MOV     #1802.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 1802.
                                ;+
                                ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR.
                                ; USE BYTE ACCESSES.
                                ;-
024764  012702  000017          MOV     #17,R2          ;SET LOOP COUNT.
024770  013704  002214          MOV     CSRA,R4         ;GET CSR ADDRESS.
024774  110214                  2$:     MOVB    R2,(R4) ;WRITE COUNT TO CSR.
024776  111401                  MOVB    (R4),R1         ;READ BACK THE CONTENTS OF THE CSR
025000  042701  177760          BIC     #177760,R1      ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
025004  020102                  CMP     R1,R2           ;CHECK FOR CORRECT DATA WRITTEN/READ.
025006  001406                  BEQ     4$              ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                ;REPORT 'BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D).'
025010  012737  012752  002672  MOV     #ER1601,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
025016  005003                  CLR     R3              ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
025020  005005                  CLR     R5              ;CAUSE REPORT OF LINE 0.
025022                          ERROR                   ;  >>>>> ERROR # 1802 <<<<<
025022  104460                                                                  TRAP    C$ERROR
025024  005302                  4$:     DEC     R2      ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
025026  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 <<<<<.
```

```
6215                                    ;-
6216  025030  005237  002666                 INC     ERRNBR          ;SET THE ERROR NUMBER TO 1803.
6217  025034  012703  177777                 MOV     #-1,R3          ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6218  025040  012704  000002                 MOV     #2,R4           ;INDICATE R/W ACCESS, CLEAR FIRST.
6219  025044  004737  015046                JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6220                                    ;+
6221                                    ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6222                                    ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
6223                                    ;   EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6224                                    ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1806 - 1808 <<<<<.
6225                                    ;-
6226  025050  012737  003416  002666          MOV     #1806.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6227  025056  005403                          NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6228  025060  004737  015046                JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6229                                    ;+
6230                                    ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6231                                    ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
6232                                    ;   EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6233                                    ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1809 - 1811 <<<<<.
6234                                    ;-
6235  025064  012737  003421  002666          MOV     #1809.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6236  025072  005403                          NEG     R3              ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6237  025074  005404                          NEG     R4              ;INDICATE R/W ACCESS, SET FIRST.
6238  025076  004737  015046                JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6239                                    ;+
6240                                    ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6241                                    ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
6242                                    ;   EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6243                                    ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1812 - 1814 <<<<<.
6244                                    ;-
6245  025102  012737  003424  002666          MOV     #1812.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6246  025110  005403                          NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6247  025112  004737  015046                JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6248                                    ;+
6249                                    ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6250                                    ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1815 <<<<<
6251                                    ;-
6252  025116  012737  003427  002666          MOV     #1815.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
6253  025124  004737  015256                JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
6254  025130  005037  002310          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
6255  025134                                  ENDTST
6256  025134                                                                          L10043:
6257  025134  104401                                                                          TRAP    C$ETST
```

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

                                        BGNTST
                                                                                T18::
                                         SETPRI #PRI05          ;ALLOW THE LTC TO INTERRUPT.
025136  012700  000240                                                    MOV     #PRI05,R0
025142  104441                                                          TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025144  012737  000022  002274          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (19)
025152  012737  177777  002310          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
025160  012737  000001  002664          MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
025166  012737  003555  002666          MOV     #1901.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
025174  012737  011161  002670          MOV     #EM1901,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
025202  005037  002420                  CLR     ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
025206  012700  002422                  MOV     #ERCNTB,R0
025212  004737  014026                  JSR     PC,CLR16W       ;CLEAR THE ERROR COUNTER TABLE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1901 <<<<<.
                                ;-
025216  004737  015304                  JSR     PC,RESETT       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
025222  103402                          BCS     .+6             ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
025224  000137  025406                  JMP     60$             ;YES,  EXIT THE TEST.
025230  012737  003556  002666          MOV     #1902.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 1902.
                                ;+
                                ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR.
                                ; USE BYTE ACCESSES.
                                ;-
025236  012702  000017                  MOV     #17,R2          ;SET LOOP COUNT.
025242  013704  002214                  MOV     CSRA,R4         ;GET CSR ADDRESS.
025246  142714  000017          2$:     BICB    #17,(R4)        ;CLEAR THE DUT CSR USING READ/MODIFY/WRITE.
025252  150214                          BISB    R2,(R4)         ;WRITE COUNT TO CSR USING READ/MODIFY/WRITE.
025254  111401                          MOVB    (R4),R1         ;READ BACK THE CONTENTS OF THE CSR
025256  042701  177760                  BIC     #177760,R1      ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
025262  020102                          CMP     R1,R2           ;CHECK FOR CORRECT DATA WRITTEN/READ.
025264  001406                          BEQ     4$              ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                    ;REPORT 'BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
025266  012737  012752  002672          MOV     #ER1601,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
025274  005003                          CLR     R3              ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
025276  005005                          CLR     R5              ;CAUSE REPORT OF LINE 0.
025300                                  ERROR                   ;  >>>>> ERROR # 1902 <<<<<
025300  104460                                                          TRAP    C$ERROR
025302  005302                  4$:     DEC     R2              ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
025304  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 <<<<<.
                                ;-
025306  005237  002666                INC     ERRNBR          ;SET THE ERROR NUMBER TO 1903.
025312  012703  177777                MOV     #-1,R3          ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
025316  012704  000001                MOV     #1,R4           ;INDICATE R/M/W ACCESS, CLEAR FIRST.
025322  004737  015046               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 <<<<<.
                                ;-
025326  012737  003562  002666        MOV     #1906.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
025334  005403                       NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
025336  004737  015046               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 <<<<<.
                                ;-
025342  012737  003565  002666        MOV     #1909.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
025350  005403                       NEG     R3              ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
025352  005404                       NEG     R4              ;INDICATE R/M/W ACCESS, SET FIRST.
025354  004737  015046               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 <<<<<.
                                ;-
025360  012737  003570  002666        MOV     #1912.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
025366  005403                       NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
025370  004737  015046               JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                ;+
                                ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
                                ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1915 <<<<<
                                ;-
025374  012737  003573  002666        MOV     #1915.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
025402  004737  015256               JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
025406  005037  002310        60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
025412                               ENDTST
025412                                                                  L10044:
025412  104401                                                                  TRAP    C$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 #PRI05          ;ALLOW THE LTC TO INTERRUPT.
025414  012700  000240                                                                  MOV     #PRI05,R0
025420  104441                                                                         TRAP    C$SPRI
        000023                  TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025422  012737  000023  002274  MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (20)
025430  012737  177777  002310  MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
025436  012737  000001  002664  MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
025444  012737  003721  002666  MOV     #2001.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
025452  012737  011236  002670  MOV     #EM2001,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 2001 <<<<<.
                                ;-
025460  004737  015304          JSR     PC,RESETT       ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
025464  103016                  BCC     60$             ;FATAL RESET ERROR?  YES,  EXIT THE TEST.
                                ;+
                                ; READ THE STAT REGISTER ID BIT AND VERIFY THAT IT IS CLEAR.
                                ;-
025466  017701  154530           MOV     @STATA,R1       ;READ THE STAT REGISTER CONTENTS.
025472  032701  000400           BIT     #BIT8,R1        ;CHECK THE ID BIT.
025476  001411                  BEQ     60$             ;ID BIT CLEAR?  YES, EXIT THE TEST.
025500  012737  003722  002666   MOV     #2002.,ERRNBR   ;NO,  SET THE ERROR REPORT NUMBER TO 2002.
025506  012701  011300           MOV     #EM2002,R1      ;GET THE PROPER ERROR MESSAGE.
025512  012737  012576  002672   MOV     #ER0503,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
025520                          ERROR                   ;ERROR NUMBER >>>>> 2002 <<<<<
025520  104460                                                                         TRAP    C$ERROR
025522  005037  002310  60$:     CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
025526                          ENDTST
025526                                                                          L10045:
025526  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.
                                ;*
                                ;-- ************************************************************************
025530                                  BGNTST
025530                                                                          T20::
        000024                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025530  012737  000024  002274   MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (93)
025536  012737  177777  002310   MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
025544  013702  002462           MOV    BMPCQP,R2       ;GET THE CONTENTS OF THE POINTER.
025550  012703  002464           MOV    #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
025554  020203                   CMP    R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
025556  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''

025560  012701  012121           MOV    #EM9304,R1      ;PASS THE FIRST MESSAGE TO BE REORTED.
025564                           ERRDF  9301,EM9301,ER9301 ;                    >>>>> ERROR #9301 <<<<<.
025564  104455                                                                  TRAP    C$ERDF
025566  022125                                                                  .WORD   9301
025570  012004                                                                  .WORD   EM9301
025572  013372                                                                  .WORD   ER9301

025574  012737  002464  002462           MOV    #BMPCQB,BMPCQP  ;SET POINTER BACK TO THE BEGINING OF THE QUE.

025602  005037  002310   60$:    CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
025606                           ENDTST
025606                                                                  L10046:
025606  104401                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE PARAMETER CODING SECTION



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

025610                                  BGNHRD
025610  000011                                                  .WORD L10047-L$HARD/2
025612                                                  L$HARD::

                                ;DEVICE CSR ADDRESS QUESTION:
025612                                  GPRMA   HWPTQ1,0,0,160000,177776,YES
025612  000031                                                  .WORD   T$CODE
025614  025634                                                  .WORD   HWPTQ1
025616  160000                                                  .WORD   T$LOLIM
025620  177776                                                  .WORD   T$HILIM

                                ;ACTIVE LINES BIT MAP QUESTION:
025622                                  GPRMD   HWPTQ3,2,0,MAPLNS,0,MAPLNS,YES
025622  001032                                                  .WORD   T$CODE
025624  025652                                                  .WORD   HWPTQ3
025626  000377                                                  .WORD   MAPLNS
025630  000000                                                  .WORD   T$LOLIM
025632  000377                                                  .WORD   T$HILIM
025634                                  ENDHRD
                                                                .EVEN
025634                                                  L10047:


025634  051503  020122  042101        HWPTQ1: .ASCIZ  /CSR ADDRESS: /
025642  051104  051505  035123
025650  000040
025652  041501  044524  042526        HWPTQ3: .ASCIZ  /ACTIVE LINE BIT MAP: /
025660  046040  047111  020105
025666  044502  020124  040515
025674  035120  000040
                                .EVEN
```

```
6481
6482
6483                                    .SBTTL  SOFTWARE PARAMETER CODING SECTION
6484
6485                                    ;++
6486                                    ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
6487                                    ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
6488                                    ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
6489                                    ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
6490                                    ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
6491                                    ; WITH THE OPERATOR.
6492                                    ;--
6493
6494  025700                                    BGNSFT
6495  025700  000013                                                                .WORD L10050-L$SOFT/2
6496  025702                                                                  L$SOFT::
6497
6498                                    ;UNIT NUMBER PRINTOUT QUESTION:
6499  025702                                    GPRML   SWPTQ1,0,20,YES
6500  025702  000130                                                                .WORD   T$CODE
6501  025704  025730                                                                .WORD   SWPTQ1
6502  025706  000020                                                                .WORD   20
6503                                    ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
6504  025710                                    GPRMD   SWPTQ2,2,D,177777,0,177777,YES
6505  025710  001052                                                                .WORD   T$CODE
6506  025712  026004                                                                .WORD   SWPTQ2
6507  025714  177777                                                                .WORD   177777
6508  025716  000000                                                                .WORD   T$LOLIM
6509  025720  177777                                                                .WORD   T$HILIM
6510                                    ;ROM VERSION PRINTOUT ON FIRST PASS QUESTION:
6511  025722                                    GPRML   SWPTQ3,0,1,YES
6512  025722  000130                                                                .WORD   T$CODE
6513  025724  026073                                                                .WORD   SWPTQ3
6514  025726  000001                                                                .WORD   1
6515
6516                                    .EVEN
6517
6518  025730                                    ENDSFT
6519                                                                            .EVEN
6520  025730                                                                  L10050:
6521
6522
6523  025730  042522  047520  052122          SWPTQ1: .ASCIZ  /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /
6524  025736  052440  044516  020124
6525  025744  052516  041115  051105
6526  025752  040440  020123  040505
6527  025760  044103  052440  044516
6528  025766  020124  051511  052040
6529  025774  051505  042524  035104
6530  026002  000040
6531  026004  052516  041115  051105          SWPTQ2: .ASCIZ  /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /
6532  026012  047440  020106  047111
6533  026020  044504  044526  052504
6534  026026  046101  042040  052101
6535  026034  020101  051105  047522
6536  026042  051522  052040  020117
```

```
 6537  026050  042522  047520  052122
 6538  026056  047440  020116  020101
 6539  026064  044514  042516  020072
 6540  026072     000
 6541  026073     122  046517  053040          SWPTQ3: .ASCIZ  /ROM VERSION PRINTOUT ON THE FIRST PASS: /
 6542  026100  051105  044523  047117
 6543  026106  050040  044522  052116
 6544  026114  052517  020124  047117
 6545  026122  052040  042510  043040
 6546  026130  051111  052123  050040
 6547  026136  051501  035123  000040
 6548                                          .EVEN
```

```
6549
6550
6551  026144                          $PATCH::
6552  026144  000024                          .BLKW   24
6553
6554
6555
6556
6557
6558  026214                          LASTAD
6559                                                          .EVEN
6560  026214  000000                                          .WORD   0
6561  026216  000000                                          .WORD   0
6562  026220                          L$LAST::
6563  026220                          ENDMOD
6564
6565
6566
6567
6568
6569
6570
6571          000001                          .END
```

```
ACTLNS  002210 G          1307#   3457    4330    4638*
ADR   = 000020 G          1281#
ADRPTR  014010 G          3040#   4450
ASSEMB= 000010            1014
BCOUNT  002330 G          1369#   4420    4562*
BITTBL  002346 G          1384#   3111    3457    4266    4330
BIT0  = 000001 G          1254#   2366    3542    3901    5971
BIT00 = 000001 G          1243#   1254
BIT01 = 000002 G          1242#   1253
BIT02 = 000004 G          1241#   1252
BIT03 = 000010 G          1240#   1251
BIT04 = 000020 G          1239#   1250
BIT05 = 000040 G          1238#   1249    3678    4021    4926    5006    5095    5164    5232    5311    5387    5464    5537
                          5615    5697    5796    5893
BIT06 = 000100 G          1237#   1248
BIT07 = 000200 G          1236#   1247
BIT08 = 000400 G          1235#   1246
BIT09 = 001000 G          1234#   1245
BIT1  = 000002 G          1253#   2374    5840    5939
BIT10 = 002000 G          1233#
BIT11 = 004000 G          1232#
BIT12 = 010000 G          1231#
BIT13 = 020000 G          1230#   5630    5648    5721    5823
BIT14 = 040000 G          1229#
BIT15 = 100000 G          1228#   4453
BIT2  = 000004 G          1252#
BIT3  = 000010 G          1251#
BIT4  = 000020 G          1250#   4666
BIT5  = 000040 G          1249#
BIT6  = 000100 G          1248#   4577
BIT7  = 000200 G          1247#   3831    3912    5242    5251
BIT8  = 000400 G          1246#   6388
BIT9  = 001000 G          1245#
BMPCQB  002464 G          1421#   2850    4656    5486    6415    6430
BMPCQE  002664 G          1422#   2863    3986    4657
BMPCQP  002462 G          1420#   3981    3989*   4658*   5485    6414    6430*
BOE   = 000400 G          1285#
CALMSL  013552 G          2930#   4588
CKTRAP  013776 G          3036#   4580    4601    4836    4843    4854    4859
CLKBRL  002314 G          1363#   4545*
CLKCSR  002312 G          1362#   4544*   4579
CLKHRZ  002320 G          1365#   2951*   4547*   4548    4560    4583*
CLKINT  016744 G          4412#   4555
CLKVEC  002316 G          1364#   4546*   4556
CLR16W  014026 G          3063#   6024    6102    6181    6280
CNTERR  014050 G          3095#   3485
CSRA    002214 G          1314#   3454*   3684    3697*   4026    4328*   4637*   4643    4834    4842    4851    4928    5008
                          5097    5166    5234    5313    5389    5466    5539    5617    5632    5699    5798    5815    5895
                          6038    6116    6196    6295
CSRO  = 000000 G          1208#
CTRLCF  002310 G          1357#   4679*   4721    4816*   4891*   4900*   4915*   4980*   4995*   5067*   5086*   5136*   5155*
                          5204*   5223*   5283*   5302*   5360*   5379*   5433*   5456*   5507*   5528*   5586*   5606*   5669*
                          5688*   5765*   5786*   5862*   5884*   5998*   6018*   6076*   6096*   6155*   6175*   6254*   6274*
                          6354*   6373*   6395*   6413*   6432*
C$AU  = 000052            1014#   4796
C$AUTO= 000061            1014#   4708
```

```
C$BRK = 000022          1014#  3318  4423
C$BSEG= 000004          1014#
C$BSUB= 000002          1014#
C$CEFG= 000045          1014#
C$CLCK= 000062          1014#  4542
C$CLEA= 000012          1014#  4735
C$CLOS= 000035          1014#
C$CLP1= 000006          1014#
C$CVEC= 000036          1014#
C$DCLN= 000044          1014#  4893
C$DODU= 000051          1014#  4890
C$DRPT= 000024          1014#
C$DU  = 000053          1014#  4774
C$EDIT= 000003          1014#  1079
C$ERDF= 000055          1014#  4883  5495  6425
C$ERHR= 000056          1014#
C$ERRO= 000060          1014#  3489  3646  3717  3815  3854  3868  3932  3950  4115  4961  4968  4975
                        5048   5055  5062  5124  5131  5192  5199  5264  5271  5278  5341  5348  5355
                        5420   5428  5570  5577  5660  5727  5743  5750  5756  5854  5989  6049  6128
                        6207   6307  6394
C$ERSF= 000054          1014#  3306
C$ERSO= 000057          1014#
C$ESCA= 000010          1014#
C$ESEG= 000005          1014#
C$ESUB= 000003          1014#
C$ETST= 000001          1014#  4903  4983  5070  5139  5207  5286  5363  5436  5510  5589  5672  5768
                        5865   6001  6079  6158  6257  6357  6398  6435
C$EXIT= 000032          1014#  4727
C$GETB= 000026          1014#
C$GETW= 000027          1014#
C$GMAN= 000043          1014#
C$GPHR= 000042          1014#  4629
C$GPLO= 000030          1014#
C$GPRI= 000040          1014#
C$INIT= 000011          1014#  4689
C$INLP= 000020          1014#
C$MANI= 000050          1014#
C$MEM = 000031          1014#
C$MSG = 000023          1014#  2392  2471  2503  2569  2622  2670  2723  2768  2817  2894
C$OPEN= 000034          1014#
C$PNTB= 000014          1014#  2372  2380  2456  2463  2498  2535  2604  2654  2663  2700  2755  2804
                        2848   5981
C$PNTF= 000017          1014#  3315  4675  4752
C$PNTS= 000016          1014#
C$PNTX= 000015          1014#  2386  2564  2611  2618  2714  2763  2812  2874  2886
C$QIO = 000377          1014#
C$RDBU= 000007          1014#
C$REFG= 000047          1014#  4512  4518  4524  4530
C$RESE= 000033          1014#  4536  4609  4724  4936  5016
C$REVI= 000003          1014#  1078
C$RFLA= 000021          1014#
C$RPT = 000025          1014#  4475
C$SEFG= 000046          1014#
C$SPRI= 000041          1014#  4565  4685  4918  4979  4998  5066  5083  5135  5152  5203  5220  5282
                        5299   5359  5376  5432  5453  5506  5525  5585  5603  5668  5685  5764  5783
                        5861   5881  5997  6015  6093  6172  6271  6370
```

```
C$SVEC= 000037            1014#   4558
C$TPRI= 000013            1014#
DELAY   014126 G          3134#   4017    5479    5807
DFPTBL  002176 G          1165#
DIAGMC= 000000            1014
DRADRT  002214 G          1313#   3466    4339
DROP    017712            4749    4756#
DR00MG  004507 G          1339    1805#
DR02MG  004513 G          1340    1806#
DR04MG  004520 G          1341    1807#
DR06MG  004524 G          1342    1808#
DR10MG  004531 G          1343    1809#
DR12MG  004540 G          1344    1811#
DR14MG  004551 G          1345    1813#
DR16MG  004562 G          1346    1815#
EDROP   017770            4754    4765#
EF.CON= 000036 G          1261#   4529
EF.NEW= 000035 G          1262#   4523
EF.PWR= 000034 G          1263#
EF.RES= 000037 G          1260#   4517
EF.STA= 000040 G          1259#   4511
EF0503  003030 G          1658#   2453    2460    2495    2532    2651    2697    2845
EF1401  003035 G          1659#   5978
EF1402  003137 G          1671#   2561
EF1601  003174 G          1676#   2660
EF1602  003220 G          1680#   2608
EF1603  003262 G          1686#   2615
EF1604  003324 G          1692#   2601
EF9001  003421 G          1703#
EF9002  003503 G          1712#
EF9003  003562 G          1720#
EF9004  003616 G          1725#
EF9005  003653 G          1730#
EF9006  003704 G          1735#
EF9010  003730 G          1739#   2711
EF9016  004027 G          1750#   2801
EF9017  004124 G          1761#   2760    2809
EF9018  004200 G          1769#   2752
EF9019  004305 G          1781#
EF9301  004331 G          1785#   2883
EF9302  004407 G          1793#   2871
EM0101  014362 G          3308    3324#
EM0102  014446 G          3312    3333#
EM0103  004572 G          1817#   4885
EM0201  004630 G          1822#   4920
EM0202  004676 G          1829#   4959    5046    5122    5190    5262    5339    5418
EM0203  005051 G          1848#   4966
EM0204  005214 G          1866#   4973
EM0301  005373 G          1885#   5000
EM0302  005436 G          1891#   5053
EM0303  005576 G          1908#   5060
EM0401  005755 G          1928#   5088
EM0402  006024 G          1935#   5129
EM0501  006174 G          1953#   5157
EM0502  006242 G          1960#   5197
EM0601  006416 G          1979#   5225
```

EM0602	006452	G	1984#	5269										
EM0603	006632	G	2004#	5276										
EM0701	007015	G	2024#	5304										
EM0702	007052	G	2029#	5346										
EM0703	007232	G	2049#	5353										
EM0801	007415	G	2069#	5381										
EM0802	007455	G	2075#	5426										
EM0901	007530	G	2083#	5497										
EM0902	007561	G	2088#	5493										
EM1001	007615	G	2093#	5530										
EM1002	007661	G	2100#	5568										
EM1003	007746	G	2109#	5575										
EM1101	010024	G	2117#	5608										
EM1201	010050	G	2121#	5690										
EM1202	010067	G	2124#	5741										
EM1203	010153	G	2133#	5754										
EM1204	010231	G	2141#	5748										
EM1205	010265	G	2146#	5658	5725									
EM1301	010311	G	2150#	5788										
EM1302	010335	G	2154#	5852										
EM1401	010374	G	2160#	5886										
EM1402	010424	G	2164#	5917										
EM1403	010512	G	2173#	5958										
EM1404	010565	G	2181#	5964										
EM1405	010637	G	2189#	2543										
EM1406	010652	G	2191#	2550										
EM1407	010665	G	2193#	2544	2551									
EM1408	010677	G	2195#	2547	2554									
EM1601	010705	G	2197#	3712										
EM1604	010770	G	2206#	6021										
EM1701	011036	G	2213#	6099										
EM1801	011113	G	2221#	6178										
EM1901	011161	G	2228#	6277										
EM2001	011236	G	2236#	6376										
EM2002	011300	G	2242#	6391										
EM9009	011353	G	2250#											
EM9010	011377	G	2254#											
EM9014	011423	G	2258#	2696										
EM9017	011517	G	2269#	3944										
EM9018	011630	G	2282#	3808										
EM9019	011640	G	2284#	3861										
EM9020	011655	G	2287#	3847										
EM9022	011701	G	2291#	3895										
EM9023	011720	G	2294#	3906										
EM9024	011742	G	2297#	3920										
EM9026	011760	G	2300#											
EM9301	012004	G	2304#	6427										
EM9302	012024	G	2307#	2851										
EM9303	012054	G	2311#	2869										
EM9304	012121	G	2318#	6423										
ENDIT	017632		4533	4679#										
ERCNTB	002422	G	1415#	2709	3101	3104*	6023	6101	6180	6279				
ERRBLK	002672	G	1429#	3438*	3640*	3713*	3809*	3848*	3862*	3881*	3947*	4113*	4921*	5001*	5089*
			5158*	5226*	5305*	5417*	5425*	5531*	5609*	5691*	5789*	5887*	5987*	6045*	6124*
			6203*	6303*	6392*										
ERRMSG	002670	G	1428#	2657	4920*	5000*	5088*	5157*	5225*	5304*	5381*	5530*	5608*	5690*	5788*

```
                    5886*  6021*  6099*  6178*  6277*  6376*
ERRNBR  002666 G    1427#  3551   3574*  3582*  3592*  3603*  3801   3807*  3845*  3846*  3859*  3860*  3879*
                    3880*  3945*  3946*  4958*  4965*  4972*  5045*  5052*  5059*  5121*  5128*  5189*  5196*
                    5261*  5268*  5275*  5338*  5345*  5352*  5416*  5424*  5458*  5501*  5560*  5567*  5574*
                    5580*  5657*  5663*  5724*  5733*  5740*  5747*  5753*  5759*  5790*  5813*  5822*  5829*
                    5850*  5916*  5959*  5965*  5992*  6020*  6036*  6057*  6066*  6074*  6098*  6114*  6136*
                    6145*  6153*  6177*  6190*  6216*  6226*  6235*  6245*  6252*  6276*  6289*  6316*  6326*
                    6335*  6345*  6352*  6375*  6390*
ERRTYP  002664 G    1426#  4919*  4999*  5087*  5156*  5224*  5303*  5380*  5457*  5529*  5607*  5689*  5787*
                    5885*  6019*  6097*  6176*  6275*  6374*
ERSMRF  002420 G    1414#  2703   3111*  3635   6022*  6100*  6179*  6278*
ER0101  012176 G    2362#  4886
ER0201  012514 G    2443#  4921   5001   5089   5158   5226   5305   5425
ER0503  012576 G    2491#  3947   5417   5531   5609   5691   5789   5887   6392
ER1401  012622 G    2528#  5987
ER1601  012752 G    2594#  3438   6045   6124   6203   6303
ER1603  013050 G    2645#  3713   4113
ER9004  013130 G    2693#  3640
ER9007  013232 G    2746#  3809   3862
ER9008  013310 G    2789#  3848   3881
ER9301  013372 G    2839#  5498   6428
EVL    = 000004 G   1279#
E$END  = 002100     1014#
E$LOAD= 000035      1014#  1102
F$AU   = 000015     1014#  4786   4795
F$AUTO= 000020      1014#  4703   4707
F$BGN  = 000040     1014#  1024   2362   2443   2491   2528   2594   2645   2693   2746   2789   2839   4465
                    4470   4485   4508   4703   4719   4727   4746   4769   4786   4790   4813   4902   4912
                    4982   4992   5069   5080   5138   5149   5206   5217   5285   5296   5362   5373   5435
                    5450   5509   5522   5588   5600   5671   5682   5767   5780   5864   5878   6000   6012
                    6078   6090   6157   6169   6256   6268   6356   6367   6397   6410   6434   6452   6495
                    6564
F$CLEA= 000007      1014#  4719   4734
F$DU   = 000016     1014#  4746   4773
F$END  = 000041     1014#  1024   2393   2472   2504   2570   2623   2671   2724   2769   2818   2895   4468
                    4476   4690   4709   4727   4736   4767   4775   4788   4797   4813   4902   4904   4912
                    4982   4984   4992   5069   5071   5080   5138   5140   5149   5206   5208   5217   5285
                    5287   5296   5362   5364   5373   5435   5437   5450   5509   5511   5522   5588   5590
                    5600   5671   5673   5682   5767   5769   5780   5864   5866   5878   6000   6002   6012
                    6078   6080   6090   6157   6159   6169   6256   6258   6268   6356   6358   6367   6397
                    6399   6410   6434   6436   6471   6521   6564
F$HARD= 000004      1014#  6452   6469
F$HW   = 000013     1014#  1163   1171
F$INIT= 000006      1014#  4508   4688
F$JMP  = 000050     1014#  4468   4727   4767   4788
F$MOD  = 000000     1014#  1024   6564
F$MSG  = 000011     1014#  2362   2391   2443   2470   2491   2502   2528   2568   2594   2621   2645   2669
                    2693   2722   2746   2767   2789   2816   2839   2893
F$PROT= 000021      1014#  4485   4492
F$PWR  = 000017     1014#
F$RPT  = 000012     1014#  4465   4474
F$SEG  = 000003     1014#
F$SOFT= 000005      1014#  6495   6519
F$SRV  = 000010     1014#
F$SUB  = 000002     1014#
F$SW   = 000014     1014#  1184   1192
```

```
F$TEST= 000001          1014#   4814    4902    4913    4982    4993    5069    5081    5138    5150    5206    5218    5285
                        5297    5362    5374    5435    5451    5509    5523    5588    5601    5671    5683    5767    5781
                        5864    5879    6000    6013    6078    6091    6157    6170    6256    6269    6356    6368    6397
                        6411    6434
GETPRM  017452          4532    4618    4622#   4633
GPRSOB  002406 G        1404#   3603    4071
G$CNTO= 000200          1014#
G$DELM= 000372          1014#
G$DISP= 000003          1014#
G$EXCP= 000400          1014#
G$HILI= 000002          1014#
G$LOLI= 000001          1014#
G$NO  = 000000          1014#
G$OFFS= 000400          1014#   6457    6463    6500    6505    6512
G$OFSI= 000376          1014#   6457    6463    6500    6505    6512
G$PRMA= 000001          1014#   6457
G$PRMD= 000002          1014#   6463    6505
G$PRML= 000000          1014#   6500    6512
G$RADA= 000140          1014#
G$RADB= 000000          1014#
G$RADD= 000040          1014#   6505
G$RADL= 000120          1014#   6500    6512
G$RADO= 000020          1014#   6457    6463
G$XFER= 000004          1014#
G$YES = 000010          1014#   6457    6463    6500    6505    6512
HELP  = 000000             1#   1007    1014    1019    1032    1120    1151    1172    1193    1196    1224    1294    1424
                        1610    1627    1639    1640    1646    1648    1649    2329    2896    4455    4477    4493    4690
                        4705    4709    4721    4730    4736    4747    4775    4787    6436    6455    6472    6481    6498
                        6523    6549    6554
HOE   = 100000 G        1292#
HWPTQ1  025634          6458    6473#
HWPTQ3  025652          6464    6476#
IBE   = 010000 G        1289#
IDU   = 000040 G        1282#
IER   = 020000 G        1290#
IESTAT  002276 G        1352#   3724*
ISR   = 000100 G        1283#
IXE   = 004000 G        1288#
I$AU  = 000041          1014#   4786#   4797#
I$AUTO= 000041          1014#   4703#   4709#
I$CLN = 000041          1014#   4719#   4727    4736#
I$DU  = 000041          1014#   4746#   4775#
I$HRD = 000041          6452#   6471#
I$INIT= 000041          1014#   4508#   4690#
I$MOD = 000041          1014#   1024#   6564#
I$MSG = 000041          1014#   2362#   2393#   2443#   2472#   2491#   2504#   2528#   2570#   2594#   2623#   2645#   2671#
                        2693#   2724#   2746#   2769#   2789#   2818#   2839#   2895#
I$PROT= 000040          1014#   4485#
I$PTAB= 000041          1014#
I$PWR = 000041          1014#
I$RPT = 000041          1014#   4465#   4476#
I$SEG = 000041          1014#   4813    4912    4992    5080    5149    5217    5296    5373    5450    5522    5600    5682
                        5780    5878    6012    6090    6169    6268    6367    6410
I$SETU= 000041          1014#
I$SFT = 000041          6495#   6521#
I$SRV = 000041          1014#
```

```
CVDHAAO DHV-11 FUNC TST PART1   MACY11 30A(1052)  12-JUL-83  10:52  PAGE 164
CVDHAA.P11    12-JUL-83 00:42          CROSS REFERENCE TABLE -- USER SYMBOLS

I$SUB = 000041            1014#   4813    4912    4992    5080    5149    5217    5296    5373    5450    5522    5600    5682
                          5780    5878    6012    6090    6169    6268    6367    6410
I$TST = 000041            1014#   4813#   4902#   4904#   4912#   4982#   4984#   4992#   5069#   5071#   5080#   5138#   5140#
                          5149#   5206#   5208#   5217#   5285#   5287#   5296#   5362#   5364#   5373#   5435#   5437#   5450#
                          5509#   5511#   5522#   5588#   5590#   5600#   5671#   5673#   5682#   5767#   5769#   5780#   5864#
                          5866#   5878#   6000#   6002#   6012#   6078#   6080#   6090#   6157#   6159#   6169#   6256#   6258#
                          6268#   6356#   6358#   6367#   6397#   6399#   6410#   6434#   6436#
J$JMP = 000167            1014#   4468    4767    4788
LNCTRA  002224 G          1318#
LNCTRO= 000010 G          1213#
LOE   = 040000 G          1291#
LOT   = 000010 G          1280#
LPCSLT= 000036 G          1476#
LPRA    002220 G          1316#
LPRO  = 000004 G          1211#   3459    4332
L$ACP   002110 G          1109#
L$APT   002036 G          1067#
L$AU    017776 G          4786#
L$AUT   002070 G          1093#
L$AUTO  017646 G          1110    4703#
L$CCP   002106 G          1107#
L$CLEA  017650 G          1108    4719#
L$CO    002032 G          1063#
L$DEPO  002011 G          1045#
L$DESC  002744 G          1100    1631#
L$DESP  002076 G          1099#
L$DEVP  002060 G          1085#
L$DISP  002124 G          1070    1129#
L$DLY   002116 G          1115#
L$DTP   002040 G          1069#
L$DTYP  002034 G          1065#
L$DU    017666 G          1096    4746#
L$DUT   002072 G          1095#
L$DVTY  002734 G          1086    1622#
L$EF    002052 G          1080#
L$ENVI  002044 G          1073#
L$ERRT  002664 G          1104    1425#
L$ETP   002102 G          1103#
L$EXP1  002046 G          1075#
L$EXP4  002064 G          1089#
L$EXP5  002066 G          1091#
L$HARD  025612 G          1052    6452    6453#
L$HIME  002120 G          1117#
L$HPCP  002016 G          1051#
L$HPTP  002022 G          1055#
L$HW    002176 G          1056    1163    1164#
L$ICP   002104 G          1105#
L$INIT  017052 G          1106    4508#
L$LADP  002026 G          1059#
L$LAST  026220 G          1060    6562#
L$LOAD  002100 G          1101#
L$LUN   002074 G          1097#
L$MREV  002050 G          1077#
L$NAME  002000 G          1034#
L$PRIO  002042 G          1071#
L$PROT  017044 G          1112    4485#
```

```
CVDHAA0 DHV-11 FUNC TST PART1   MACY11 30A(1052)  12-JUL-83  10:52  PAGE 165
CVDHAA.P11    12-JUL-83 00:42          CROSS REFERENCE TABLE -- USER SYMBOLS

L$PRT   002112 G          1111#
L$REPP  002062 G          1087#
L$REV   002010 G          1043#
L$RPT   017036 G          1088    4465#
L$SOFT  025702 G          1054    6495    6496#
L$SPC   002056 G          1083#
L$SPCP  002020 G          1053#
L$SPTP  002024 G          1057#
L$STA   002030 G          1061#
L$SW    002204 G          1058    1184    1185#
L$TEST  002114 G          1113#
L$TIML  002014 G          1049#
L$UNIT  002012 G          1047#   4624    4668
L10000  002202            1163    1171#
L10001  002210            1184    1192#
L10002  012300            2391#
L10003  012574            2470#
L10004  012620            2502#
L10005  012750            2568#
L10006  013046            2621#
L10007  013126            2669#
L10010  013230            2722#
L10011  013306            2767#
L10012  013370            2816#
L10013  013550            2893#
L10014  017042            4469    4474#
L10016  017644            4688#
L10017  017646            4707#
L10020  017664            4728    4734#
L10021  017774            4768    4773#
L10022  020002            4789    4795#
L10023  020272            4902#
L10024  020522            4982#
L10025  020766            5069#
L10026  021164            5138#
L10027  021356            5206#
L10030  021574            5285#
L10031  022002            5362#
L10032  022210            5435#
L10033  022376            5509#
L10034  022612            5588#
L10035  023040            5671#
L10036  023322            5767#
L10037  023622            5864#
L10040  024222            6000#
L10041  024440            6078#
L10042  024662            6157#
L10043  025134            6256#
L10044  025412            6356#
L10045  025526            6397#
L10046  025606            6434#
L10047  025634            6452    6470#
L10050  025730            6495    6520#
MAPLNS= 000377 G          1205#   6463    6465
MFUNIT  002772 G          1653#   4672
MMENAB  002342 G          1378#   4600*
```

```
MMPRES  002340 G          1377#   4599*   4604*
MMSR0   002336 G          1376#   4598
MSG1    012302 G          2369    2394#
MSG2    012360 G          2377    2402#
MSG3    012437 G          2383    2410#
MSLCNT  002334 G          1371#   2964    2983    3002*   3210
MSLGET  014166 G          3186#   3278    3689    3706    4268    4929    4939    4946    5009    5019    5028    5039    5100
                          5169    5237    5316    5392    5467    5540    5550    5618    5633    5644    5649    5700    5709
                          5799    5816    5896    5904
MSLOOP  014302 G          3140    3271#
MSTICK  002332 G          1370#   2996    4550*   4552*
NDERPT  002206 G          1189#   3105    3107
NEWPAS  017432            4526    4606    4611#   4625
NEWRES  017424            4520    4608#
NEWSTA  017116            4514    4534#
NUMLNS= 000010 G          1204#   3501    3873    4021    4383    4871    5408    5913
OOPS    014316 G          2975    3001    3142    3302#
OPTION  002204 G          1188#   4666    5971
O$APTS= 000000            1014#   1061
O$AU  = 000000            1014#   1093
O$BGNR= 000001            1014#   1031#   1087
O$BGNS= 000001            1014#   1031#   1053
O$DU  = 000001            1014#   1031#   1095
O$ERRT= 000001            1014#   1031#   1103
O$GNSW= 000001            1014#   1031#   1057
O$POIN= 000001            1014#   1031#   1119
O$SETU= 000000            1014#   1047    6560
PAR0A   002344 G          1379#
PASCNT  002300 G          1353#   4605*   4610*   4617*   4619*   5973
PCSLOT= 000016 G          1477#
PNT   = 001000 G          1286#
PREGRT  002716 G          1600#
PREG05  002674            1581#   2364    2445    2647    2841    2931    3037    3064    3096    3135    3187    3272    3303
                          3365    3437    3536    3634    3677    3752    3795    3980    4015    4111    4154    4257    4318
PRI   = 002000 G          1287#
PRI00 = 000000 G          1275#
PRI01 = 000040 G          1274#
PRI02 = 000100 G          1273#
PRI03 = 000140 G          1272#
PRI04 = 000200 G          1271#
PRI05 = 000240 G          1270#   4564    4917    4997    5082    5151    5219    5298    5375    5452    5524    5602    5684
                          5782    5880    6014    6092    6171    6270    6369
PRI06 = 000300 G          1269#   1363    4554
PRI07 = 000340 G          1072    1268#   4684    4978    5065    5134    5202    5281    5358    5431    5505    5584    5667
                          5763    5860    5996
PUFIFO  014544 G          3364#   5480
RBUFA   002216 G          1315#   3367    3802    3882    4647    5833    5919
RBUF0 = 000002 G          1209#
RDPDR   014626 G          3436#   3575    3583    3593
REGTST  015046 G          3535#   6060    6069    6139    6148    6219    6228    6238    6247    6319    6328    6338    6347
REPSMR  015256 G          3633#   6075    6154    6253    6353
RESETT  015304 G          3676#   6030    6108    6187    6286    6382
RMATBB  002254 G          1339#   2596
ROLDAP  015416 G          3491    3499    3599    3751#   4372    4380
RSTRPT  015450 G          3794#   5561    5734
RXBDTX= 000030 G          1220#
```

```
RXBETX= 000020 G  1219#
RXBFUL= 000100 G  1221#
R0SLOT= 000002 G  1483#  3228*
R1SLOT= 000004 G  1482#  3229*  4224*
R2SLOT= 000006 G  1481#  3761*  4272*
R3SLOT= 000010 G  1480#  3467  4340  4344
R4SLOT= 000012 G  1479#  3555  3560  4349  4361
R5SLOT= 000014 G  1478#  1588
SAVBMP  016032 G  3384  3840  3925  3979#  5927
SFPTBL  002204 G  1186#
SKPSTS  016100 G  3698  4014#  4937  5017  5027  5099  5168  5236  5315  5391  5473  5549  5624
                  5902
STATA   002222 G  1317#  6387
STATO = 000006 G  1212#
SVCGBL= 000000    1014#  1017#  1034  1043  1045  1047  1049  1051  1053  1055  1057  1059  1061
                  1063  1065  1067  1069  1071  1073  1075  1077  1080  1083  1085  1087  1089
                  1091  1093  1095  1097  1099  1101  1103  1105  1107  1109  1111  1113  1115
                  1117  1129  1164  1165  1185  1186  1425  1622  1631  2362  2443  2491  2528
                  2594  2645  2693  2746  2789  2839  4465  4485  4508  4703  4719  4746  4786
                  6453  6496  6562#  6563
SVCINS= 000001    1014#  1035  1036  1037  1038  1039  1040  1041  1042  1044  1046  1048  1050
                  1052  1054  1056  1058  1060  1062  1064  1066  1068  1070  1072  1074  1076
                  1078  1079  1081  1082  1084  1086  1088  1090  1092  1094  1096  1098  1100
                  1102  1104  1106  1108  1110  1112  1114  1116  1118  1128  1130  1131  1132
                  1133  1134  1135  1136  1137  1138  1139  1140  1141  1142  1143  1144  1145
                  1146  1147  1148  1149  1163  1184  1623  1625  1632  1636  2369  2370  2371
                  2372  2373  2377  2378  2379  2380  2381  2383  2384  2385  2386  2387  2392
                  2452  2453  2454  2455  2456  2457  2459  2460  2461  2462  2463  2464  2471
                  2494  2495  2496  2497  2498  2499  2503  2531  2532  2533  2534  2535  2536
                  2559  2560  2561  2562  2563  2564  2565  2569  2599  2600  2601  2602  2603
                  2604  2605  2607  2608  2609  2610  2611  2612  2614  2615  2616  2617  2618
                  2619  2622  2650  2651  2652  2653  2654  2655  2659  2660  2661  2662  2663
                  2664  2670  2696  2697  2698  2699  2700  2701  2709  2710  2711  2712  2713
                  2714  2715  2723  2750  2751  2752  2753  2754  2755  2756  2758  2759  2760
                  2761  2762  2763  2764  2768  2799  2800  2801  2802  2803  2804  2805  2807
                  2808  2809  2810  2811  2812  2813  2817  2844  2845  2846  2847  2848  2849
                  2871  2872  2873  2874  2875  2880  2881  2882  2883  2884  2885  2886  2887
                  2894  3306  3307  3308  3309  3312  3313  3314  3315  3316  3318  3489  3646
                  3717  3815  3854  3868  3932  3950  4115  4423  4468  4469  4475  4511  4512
                  4514  4517  4518  4520  4523  4524  4526  4529  4530  4532  4536  4541  4542
                  4543  4554  4555  4556  4557  4558  4559  4564  4565  4609  4628  4629  4630
                  4632  4671  4672  4673  4674  4675  4676  4684  4685  4689  4708  4724  4727
                  4728  4735  4748  4749  4750  4751  4752  4753  4767  4768  4774  4788  4789
                  4796  4883  4884  4885  4886  4889  4890  4893  4903  4917  4918  4936  4961
                  4968  4975  4978  4979  4983  4997  4998  5016  5048  5055  5062  5065  5066
                  5070  5082  5083  5124  5131  5134  5135  5139  5151  5152  5192  5199  5202
                  5203  5207  5219  5220  5264  5271  5278  5281  5282  5286  5298  5299  5341
                  5348  5355  5358  5359  5363  5375  5376  5420  5428  5431  5432  5436  5452
                  5453  5495  5496  5497  5498  5505  5506  5510  5524  5525  5570  5577  5584
                  5585  5589  5602  5603  5660  5667  5668  5672  5684  5685  5727  5743  5750
                  5756  5763  5764  5768  5782  5783  5854  5860  5861  5865  5880  5881  5976
                  5977  5978  5979  5980  5981  5982  5989  5996  5997  6001  6014  6015  6049
                  6079  6092  6093  6128  6158  6171  6172  6207  6257  6270  6271  6307  6357
                  6369  6370  6394  6398  6425  6426  6427  6428  6435  6452  6457  6458  6459
                  6460  6463  6464  6465  6466  6467  6469  6495  6500  6501  6502  6505  6506
                  6507  6508  6509  6512  6513  6514  6519  6559  6560  6561
```

```
SVCSUB= 000001          1014# 1016#
SVCTAG= 000001          1014# 1018# 1171 1192 2391 2470 2502 2568 2621 2669 2722 2767 2816
                        2893  4474  4688 4707 4734 4773 4795 4902 4982 5069 5138 5206 5285
                        5362  5435  5509 5588 5671 5767 5864 6000 6078 6157 6256 6356 6397
                        6434  6470  6520
SVCTST= 000001          1014# 1015# 4813 4912 4992 5080 5149 5217 5296 5373 5450 5522 5600
                        5682  5780  5878 6012 6090 6169 6268 6367 6410
SWAP0   016156 G        3550  3563  3572 3576 4059#
SWPTQ1  025730          6501  6523#
SWPTQ2  026004          6506  6531#
SWPTQ3  026073          6513  6541#
S$LSYM= 010000          1014# 1172# 1193# 2392# 2471# 2503# 2569# 2622# 2670# 2723# 2768# 2817# 2894#
                        4475# 4689# 4708# 4735# 4774# 4796# 4903# 4983# 5070# 5139# 5207# 5286# 5363#
                        5436# 5510# 5589# 5672# 5768# 5865# 6001# 6079# 6158# 6257# 6357# 6398# 6435#
                        6471# 6521#
TIMER1  002322 G        1366# 2940* 2941 2958 4412 4414*
TIMER2  002324 G        1367# 4415  4417*
TIMER3  002326 G        1368# 4418* 4420*
TNUM  = 000024 G        4814# 4815  4913# 4914 4993# 4994 5084# 5085 5153# 5154 5221# 5222 5300#
                        5301  5377# 5378  5454# 5455 5526# 5527 5604# 5605 5686# 5687 5784# 5785
                        5882# 5883  6016# 6017 6094# 6095 6173# 6174 6272# 6273 6371# 6372 6411#
                        6412
TP4FLG  002304 G        1355# 3038* 3040 4453* 4576* 4595* 4833*
TP4RTN  017014 G        4450# 4572  4594 4821
TP4VEC  002302 G        1354# 4452  4571* 4581 4593* 4602 4820* 4878
TSABRT  016226 G        4110# 5502  5581 5664 5760 5857 5993
TSTNUM  002274 G        1351# 3982  4815* 4914* 4994* 5085* 5154* 5222* 5301* 5378* 5455* 5527* 5605*
                        5687* 5785* 5883* 6017* 6095* 6174* 6273* 6372* 6412*
TXAD1A  002226 G        1319#
TXAD10= 000012 G        1214#
TXAD2A  002230 G        1320#
TXAD20= 000014 G        1215#
TXBFCA  002232 G        1321# 4029  5808*
TXBFCO= 000016 G        1216# 3493  4374
TXCHA   002216 G        1315#
TXCHR0= 000002 G        1210#
T$ARGC= 000003          1035# 1036# 1037# 1038# 1039# 1040# 2369# 2373 2377# 2381 2383# 2387 2452#
                        2457  2459# 2464 2494# 2499 2531# 2536 2559# 2565 2599# 2605 2607# 2612
                        2614# 2619  2650# 2655 2659# 2664 2696# 2701 2709# 2715 2750# 2756 2758#
                        2764  2799# 2805 2807# 2813 2844# 2849 2871# 2875 2880# 2887 3312# 3316
                        4671# 4676  4748# 4753 5976# 5982
T$CODE= 000130          6457# 6463# 6500# 6505# 6512#
T$ERRN= 022125          1014# 3307# 4884# 5496# 6426#
T$EXCP= 000000          6457# 6461  6463# 6468 6505# 6510
T$FLAG= 000050          4468# 4470  4727# 4767# 4769 4788# 4790
T$GMAN= 000000          1014#
T$HILI= 177777          6457# 6460  6463# 6467 6505# 6509
T$LAST= 000001          1014# 6560#
T$LOLI= 000000          6457# 6459  6463# 6466 6505# 6508
T$LSYM= 010000          1014# 1172  1193  2392 2471 2503 2569 2622 2670 2723 2768 2817 2894
                        4475  4689  4708  4735 4774 4796 4903 4983 5070 5139 5207 5286 5363
                        5436  5510  5589  5672 5768 5865 6001 6079 6158 6257 6357 6398 6435
                        6471  6521
T$LTNO= 000024          6563#
T$NEST= 177777          1014# 1024# 1163# 1171# 1184# 1192# 2362# 2391# 2443# 2470# 2491# 2502# 2528#
                        2568# 2594# 2621# 2645# 2669# 2693# 2722# 2746# 2767# 2789# 2816# 2839# 2893#
```

```
                    4465#  4474#  4485#  4492#  4508#  4688#  4703#  4707#  4719#  4734#  4746#  4773#  4786#
                    4795#  4814#  4902#  4913#  4982#  4993#  5069#  5081#  5138#  5150#  5206#  5218#  5285#
                    5297#  5362#  5374#  5435#  5451#  5509#  5523#  5588#  5601#  5671#  5683#  5767#  5781#
                    5864#  5879#  6000#  6013#  6078#  6091#  6157#  6170#  6256#  6269#  6356#  6368#  6397#
                    6411#  6434#  6452#  6469#  6495#  6519#  6564#
T$NS0 = 000000      1024#  6564
T$NS1 = 000005      1163#  1171   1184#  1192   2362#  2391   2443#  2470   2491#  2502   2528#  2568   2594#
                    2621   2645#  2669   2693#  2722   2746#  2767   2789#  2816   2839#  2893   4465#  4474
                    4485#  4492   4508#  4688   4703#  4707   4719#  4734   4746#  4773   4786#  4795   4814#
                    4902   4913#  4982   4993#  5069   5081#  5138   5150#  5206   5218#  5285   5297#  5362
                    5374#  5435   5451#  5509   5523#  5588   5601#  5671   5683#  5767   5781#  5864   5879#
                    6000   6013#  6078   6091#  6157   6170#  6256   6269#  6356   6368#  6397   6411#  6434
                    6452#  6469   6495#  6519
T$PTNU= 000000      1014#
T$SAVL= 177777      1014#
T$SEGL= 177777      1014#
T$SUBN= 000000      1014#  4813#  4912#  4992#  5080#  5149#  5217#  5296#  5373#  5450#  5522#  5600#  5682#
                    5780#  5878#  6012#  6090#  6169#  6268#  6367#  6410#
T$TAGL= 177777      1014#
T$TAGN= 010051      1014#  1163#  1184#  2362#  2443#  2491#  2528#  2594#  2645#  2693#  2746#  2789#  2839#
                    4465#  4485#  4508#  4703#  4719#  4746#  4786#  4814#  4913#  4993#  5081#  5150#  5218#
                    5297#  5374#  5451#  5523#  5601#  5683#  5781#  5879#  6013#  6091#  6170#  6269#  6368#
                    6411#  6452#  6495#
T$TEMP= 000000      1130#  1131#  1132#  1133#  1134#  1135#  1136#  1137#  1138#  1139#  1140#  1141#  1142#
                    1143#  1144#  1145#  1146#  1147#  1148#  1149#  1150#  1171#  1192#  2391#  2470#  2502#
                    2568#  2621#  2669#  2722#  2767#  2816#  2893#  4468#  4469   4474#  4492#  4688#  4707#
                    4727#  4728   4734#  4767#  4768   4773#  4788#  4789   4795#  4902#  4982#  5069#  5138#
                    5206#  5285#  5362#  5435#  5509#  5588#  5671#  5767#  5864#  6000#  6078#  6157#  6256#
                    6356#  6397#  6434#  6457#  6463#  6469#  6500#  6505#  6512#  6519#  6564#
T$TEST= 000024      1014#  4813#  4912#  4992#  5080#  5149#  5217#  5296#  5373#  5450#  5522#  5600#  5682#
                    5780#  5878#  6012#  6090#  6169#  6268#  6367#  6410#  6563
T$TSTM= 177777      1014#  2372   2380   2386   2392   2456   2463   2471   2498   2503   2535   2564   2569
                    2604   2611   2618   2622   2654   2663   2670   2700   2714   2723   2755   2763   2768
                    2804   2812   2817   2848   2874   2886   2894   3306   3315   3318   3489   3646   3717
                    3815   3854   3868   3932   3950   4115   4423   4475   4512   4518   4524   4530   4536
                    4542   4558   4565   4609   4629   4675   4685   4689   4708   4724   4727   4735   4752
                    4774   4796   4883   4890   4893   4903   4918   4936   4961   4968   4975   4979   4983
                    4998   5016   5048   5055   5062   5066   5070   5083   5124   5131   5135   5139   5152
                    5192   5199   5203   5207   5220   5264   5271   5278   5282   5286   5299   5341   5348
                    5355   5359   5363   5376   5420   5428   5432   5436   5453   5495   5506   5510   5525
                    5570   5577   5585   5589   5603   5660   5668   5672   5685   5727   5743   5750   5756
                    5764   5768   5783   5854   5861   5865   5881   5981   5989   5997   6001   6015   6049
                    6079   6093   6128   6158   6172   6207   6257   6271   6307   6357   6370   6394   6398
                    6425   6435
T$TSTS= 000001      1014#  4814#  4913#  4993#  5081#  5150#  5218#  5297#  5374#  5451#  5523#  5601#  5683#
                    5781#  5879#  6013#  6091#  6170#  6269#  6368#  6411#
T$$AU = 010022      4786#  4788   4795
T$$AUT= 010017      4703#  4707
T$$CLE= 010020      4719#  4727   4734
T$$DU = 010021      4746#  4767   4773
T$$HAR= 010047      6452#  6470
T$$HW = 010000      1163#  1171
T$$INI= 010016      4508#  4688
T$$MSG= 010013      2362#  2391   2443#  2470   2491#  2502   2528#  2568   2594#  2621   2645#  2669   2693#
                    2722   2746#  2767   2789#  2816   2839#  2893
T$$PRO= 010015      4485#
```

CVDHAA0 DHV-11 FUNC TST PART1 MACY11 30A(1052) 12-JUL-83 10:52 PAGE 170
CVDHAA.P11 12-JUL-83 00:42 CROSS REFERENCE TABLE -- USER SYMBOLS

```
T$$RPT= 010014          4465#   4468    4474
T$$SOF= 010050          6495#   6520
T$$SW = 010001          1184#   1192
T$$TES= 010046          4814#   4902    4913#   4982    4993#   5069    5081#   5138    5150#   5206    5218#   5285    5297#
                        5362    5374#   5435    5451#   5509    5523#   5588    5601#   5671    5683#   5767    5781#   5864
                        5879#   6000    6013#   6078    6091#   6157    6170#   6256    6269#   6356    6368#   6397    6411#
                        6434
T1      020004 G        1130    4813#
T10     022400 G        1139    5522#
T11     022614 G        1140    5600#
T12     023042 G        1141    5682#
T13     023324 G        1142    5780#
T14     023624 G        1143    5878#
T15     024224 G        1144    6012#
T16     024442 G        1145    6090#
T17     024664 G        1146    6169#
T18     025136 G        1147    6268#
T19     025414 G        1148    6367#
T2      020274 G        1131    4912#
T20     025530 G        1149    6410#
T3      020524 G        1132    4992#
T4      020770 G        1133    5080#
T5      021166 G        1134    5149#
T6      021360 G        1135    5217#
T7      021576 G        1136    5296#
T8      022004 G        1137    5373#
T9      022212 G        1138    5450#
UAM   = 000200 G        1284#
UNBTTB  002234 G        1327#   3465    3482    4338
UNITN   002212 G        1308#   4612*   4623*   4624    4628    4671    4889
UNSDIV  016340 G        2999    4153#
WAIBIS  016474 G        4256#
WDPDR   016550 G        3573    3581    4317#
WORD1   002306 G        1356#   4577*   4578    4596*   4597
X$ALWA= 000000          1014#
X$FALS= 000040          1014#
X$OFFS= 000400          1014#
X$TRUE= 000020          1014#
$PATCH  026144 G        6551#
.     = 026220          1021#   1415#   1421#   1625#   2326#   4469    4728    4764#   4768    4789    6031    6109    6188
                        6287    6552#
```

```
BCOMPL      1#   1014#  4513  4519  4525  4631
BERROR      1#   1014#
BGNAU       1#   1014#  4785
BGNAUT      1#   1014#  4702
BGNCLN      1#   1014#  4718
BGNDU       1#   1014#  4745
BGNHRD      1#   1014#  6451
BGNHW       1#   1014#  1162
BGNINI      1#   1014#  4507
BGNMOD      1#   1014#  1023
BGNMSG      1#   1014#  2361  2442  2490  2527  2593  2644  2692  2745  2788  2838
BGNPRO      1#   1014#  4484
BGNPTA      1#   1014#
BGNRPT      1#   1014#  4464
BGNSEG      1#   1014#
BGNSET      1#   1014#
BGNSFT      1#   1014#  6494
BGNSRV      1#   1014#
BGNSUB      1#   1014#
BGNSW       1#   1014#  1183
BGNTST      1#   1014#  4812  4911  4991  5079  5148  5216  5295  5372  5449  5521  5599  5681  5779
         5877    6011   6089  6168  6267  6366  6409
BNCOMP      1#   1014#  4531
BNERRO      1#   1014#
BREAK       1#   1014#  3317  4422
BRESET      1#   1014#  4535  4608  4723  4935  5015
CKLOOP      1#   1014#
CLOCK       1#   1014#  4540
CLOSE       1#   1014#
CLRVEC      1#   1014#
COMMEN      1#   1014#
DELAY       1#   1014#
DESCRI      1#   1014#  1630
DEVTYP      1#   1014#  1621
DISPAT      1#   1014#  1127
DISPLA      1#   1014#
DOCLN       1#   1014#  4892
DODU        1#   1014#  4888
DORPT       1#   1014#
ENDAU       1#   1014#  4794
ENDAUT      1#   1014#  4706
ENDCLN      1#   1014#  4733
ENDCOM      1#   1014#
ENDDU       1#   1014#  4772
ENDHRD      1#   1014#  6468
ENDHW       1#   1014#  1170
ENDINI      1#   1014#  4687
ENDMOD      1#   1014#  6563
ENDMSG      1#   1014#  2390  2469  2501  2567  2620  2668  2721  2766  2815  2892
ENDPRO      1#   1014#  4491
ENDPTA      1#   1014#
ENDRPT      1#   1014#  4473
ENDSEG      1#   1014#
ENDSET      1#   1014#
ENDSFT      1#   1014#  6518
ENDSRV      1#   1014#
```

```
ENDSUB      1#   1014#
ENDSW       1#   1014#  1191
ENDTST      1#   1014#  4901  4981  5068  5137  5205  5284  5361  5434  5508  5587  5670  5766  5863
         5999    6077   6156  6255  6355  6396  6433
EQUALS      1#   1014#  1224
ERRDF       1#   1014#  4882  5494  6424
ERRHRD      1#   1014#
ERROR       1#   1014#  3488  3645  3716  3814  3853  3867  3931  3949  4114  4960  4967  4974  5047
         5054    5061   5123  5130  5191  5198  5263  5270  5277  5340  5347  5354  5419  5427  5569
         5576    5659   5726  5742  5749  5755  5853  5988  6048  6127  6206  6306  6393
ERRSF       1#   1014#  3305
ERRSOF      1#   1014#
ERRTBL      1#   1014#  1424
ESCAPE      1#   1014#
EXIT        1#   1014#  4467  4726  4766  4787
FEQUAL      1#   1014#
GETBYT      1#   1014#
GETPRI      1#   1014#
GETWOR      1#   1014#
GMANIA      1#   1014#
GMANID      1#   1014#
GMANIL      1#   1014#
GPHARD      1#   1014#  4627
GPRMA       1#   1014#  6456
GPRMD       1#   1014#  6462  6504
GPRML       1#   1014#  6499  6511
HEADER      1#   1014#  1033
INLOOP      1#   1014#
IOSETU      1#   1014#
IOSTAR      1#   1014#
KT11        1#   1014#
LASTAD      1#   1014#  6558
MANUAL      1#   1014#
MEMORY      1#   1014#
M$BYTE      1#   1014#  1034#  1040  1041  1042
M$CHEC      1#   1014#  4468#  4727#  4767#  4788#
M$CNTO      1#   1014#  6457#  6463#  6500#  6505#  6512#
M$COUN      1#   1014#  2369#  2377#  2383#  2452#  2459#  2494#  2531#  2559#  2599#  2607#  2614#  2650#  2659#
         2696#   2709#  2750#  2758#  2799#  2807#  2844#  2871#  2880#  3312#  4671#  4748#  5976#
M$DATA      1#   1014#  1034#  1043  1045  1047  1049  1051  1053  1055  1057  1059  1061  1063  1065
         1067    1069   1071  1073#  1075  1077  1080  1083  1085  1087  1089  1091  1093  1095  1097
         1099    1101   1103  1105  1107  1109  1111  1113  1115  1117  1622#  1631#
M$DECR      1#   1014#  1171#  1192#  2391#  2470#  2502#  2568#  2621#  2669#  2722#  2767#  2816#  2893#  4474#
         4492#   4688#  4707#  4734#  4773#  4795#  4902#  4982#  5069#  5138#  5206#  5285#  5362#  5435#  5509#
         5588#   5671#  5767#  5864#  6000#  6078#  6157#  6256#  6356#  6397#  6434#  6469#  6519#  6564#
M$DEFA      1#   1014#  6457#  6463#  6500#  6505#  6512#
M$ENDE      1#   1014#  1171#  1192#  2391#  2470#  2502#  2568#  2621#  2669#  2722#  2767#  2816#  2893#  4474#
         4688#   4707#  4734#  4773#  4795#  4902#  4982#  5069#  5138#  5206#  5285#  5362#  5435#  5509#  5588#
         5671#   5767#  5864#  6000#  6078#  6157#  6256#  6356#  6397#  6434#  6469#  6519#  6564#
M$ERRI      1#   1014#  3306#  4883#  5495#  6425#
M$ESCA      1#   1014#
M$ESCS      1#   1014#
M$EXCP      1#   1014#  6457#  6463#  6505#
M$EXIT      1#   1014#  4468#  4727#  4728  4767#  4788#
M$EXSE      1#   1014#  4468#  4727#  4767#  4788#
M$EXTJ      1#   1014#  4468#  4469  4727#  4767#  4768  4788#  4789
```

CVDHAA0 DHV-11 FUNC TST PART1 MACY11 30A(1052) 12-JUL-83 10:52 PAGE 174
CVDHAA.P11 12-JUL-83 00:42 CROSS REFERENCE TABLE -- MACRO NAMES

```
M$GEN      1#  1014#  1034#  1043#  1045#  1047#  1049#  1051#  1053#  1055#  1057#  1059#  1061#  1063#  1065#
        1067#  1069#  1071#  1073#  1075#  1077#  1080#  1083#  1085#  1087#  1089#  1091#  1093#  1095#  1097#
        1099#  1101#  1103#  1105#  1107#  1109#  1111#  1113#  1115#  1117#  1129#  1164#  1165#  1171#  1185#
        1186#  1192#  1425#  1622#  1631#  2362#  2391#  2443#  2470#  2491#  2502#  2528#  2568#  2594#  2621#
        2645#  2669#  2693#  2722#  2746#  2767#  2789#  2816#  2839#  2893#  4465#  4474#  4485#  4508#  4688#
        4703#  4707#  4719#  4734#  4746#  4773#  4786#  4795#  4813#  4902#  4912#  4982#  4992#  5069#  5080#
        5138#  5149#  5206#  5217#  5285#  5296#  5362#  5373#  5435#  5450#  5509#  5522#  5588#  5600#  5671#
        5682#  5767#  5780#  5864#  5878#  6000#  6012#  6078#  6090#  6157#  6169#  6256#  6268#  6356#  6367#
        6397#  6410#  6434#  6453#  6470#  6496#  6520#  6562#
M$GENB     1#  1014#
M$GETS     1#  1014#  1171#  1192#  2391#  2470#  2502#  2568#  2621#  2669#  2722#  2767#  2816#  2893#  4474#
        4492#  4688#  4707#  4734#  4773#  4795#  4902#  4982#  5069#  5138#  5206#  5285#  5362#  5435#  5509#
        5588#  5671#  5767#  5864#  6000#  6078#  6157#  6256#  6356#  6397#  6434#  6469#  6519#  6564#
M$GETT     1#  1014#  4468#  4727#  4767#  4788#
M$GNGB     1#  1014#  1024#  1034#  1043#  1045#  1047#  1049#  1051#  1053#  1055#  1057#  1059#  1061#  1063#
        1065#  1067#  1069#  1071#  1073#  1075#  1077#  1080#  1083#  1085#  1087#  1089#  1091#  1093#  1095#
        1097#  1099#  1101#  1103#  1105#  1107#  1109#  1111#  1113#  1115#  1117#  1128#  1129   1163#  1164
        1165   1184#  1185   1186   1425#  1622#  1631#  2362#  2443#  2491#  2528#  2594#  2645#  2693#  2746#
        2789#  2839#  4465#  4485#  4508#  4703#  4719#  4746#  4786#  6452#  6453   6495#  6496   6559#  6562
M$GNIN     1#  1014#  1034#  1035   1036   1037   1038   1039   1040#  1041#  1042#  1043#  1044   1045#  1046
        1047#  1048   1049#  1050   1051#  1052   1053#  1054   1055#  1056   1057#  1058   1059#  1060   1061#
        1062   1063#  1064   1065#  1066   1067#  1068   1069#  1070   1071#  1072   1073#  1074   1075#  1076
        1077#  1078   1079   1080#  1081   1082#  1083#  1084   1085#  1086   1087#  1088   1089#  1090   1091#
        1092   1093#  1094   1095#  1096   1097#  1098   1099#  1100   1101#  1102   1103#  1104   1105#  1106
        1107#  1108   1109#  1110   1111#  1112   1113#  1114   1115#  1116   1117#  1118   1128#  1130#  1131#
        1132#  1133#  1134#  1135#  1136#  1137#  1138#  1139#  1140#  1141#  1142#  1143#  1144#  1145#  1146#
        1147#  1148#  1149#  1163#  1184#  1622#  1623   1625   1631#  1632   1636   2369#  2370#  2371   2372#
        2373   2377#  2378#  2379   2380#  2381   2383#  2384#  2385   2386#  2387   2392#  2452#  2453#  2454#
        2455   2456#  2457   2459#  2460#  2461#  2462   2463#  2464   2471#  2494#  2495#  2496#  2497   2498#
        2499   2503#  2531#  2532#  2533#  2534   2535#  2536   2559#  2560#  2561#  2562#  2563   2564#  2565
        2569#  2599#  2600#  2601#  2602#  2603   2604#  2605   2607#  2608#  2609#  2610   2611#  2612   2614#
        2615#  2616#  2617   2618#  2619   2622#  2650#  2651#  2652#  2653   2654#  2655   2659#  2660#  2661#
        2662   2663#  2664   2670#  2696#  2697#  2698#  2699   2700#  2701   2709#  2710#  2711#  2712#  2713
        2714#  2715   2723#  2750#  2751#  2752#  2753#  2754   2755#  2756   2758#  2759#  2760#  2761#  2762
        2763#  2764   2768#  2799#  2800#  2801#  2802#  2803   2804#  2805   2807#  2808#  2809#  2810#  2811
        2812#  2813   2817#  2844#  2845#  2846#  2847   2848#  2849   2871#  2872#  2873   2874#  2875   2880#
        2881#  2882#  2883#  2884#  2885   2886#  2887   2894#  3306#  3307#  3308#  3309#  3312#  3313#  3314
        3315#  3316   3318#  3489#  3646#  3717#  3815#  3854#  3868#  3932#  3950#  4115#  4423#  4468#  4469#
        4475#  4511#  4512#  4514#  4517#  4518#  4520#  4523#  4524#  4526#  4529#  4530#  4532#  4536#  4541#
        4542#  4543#  4554#  4555#  4556#  4557#  4558#  4559   4564#  4565#  4609#  4628#  4629#  4630#  4632#
        4671#  4672#  4673#  4674   4675#  4676   4684#  4685#  4689#  4708#  4724#  4727#  4728#  4735#  4748#
        4749#  4750#  4751   4752#  4753   4767#  4768#  4774#  4788#  4789#  4796#  4883#  4884#  4885#  4886#
        4889#  4890#  4893#  4903#  4917#  4918#  4936#  4961#  4968#  4975#  4978#  4979#  4983#  4997#  4998#
        5016#  5048#  5055#  5062#  5065#  5066#  5070#  5082#  5083#  5124#  5131#  5134#  5135#  5139#  5151#
        5152#  5192#  5199#  5202#  5203#  5207#  5219#  5220#  5264#  5271#  5278#  5281#  5282#  5286#  5298#
        5299#  5341#  5348#  5355#  5358#  5359#  5363#  5375#  5376#  5420#  5428#  5431#  5432#  5436#  5452#
        5453#  5495#  5496#  5497#  5498#  5505#  5506#  5510#  5524#  5525#  5570#  5577#  5584#  5585#  5589#
        5602#  5603#  5660#  5667#  5668#  5672#  5684#  5685#  5727#  5743#  5750#  5756#  5763#  5764#  5768#
        5782#  5783#  5854#  5860#  5861#  5865#  5880#  5881#  5976#  5977#  5978#  5979#  5980   5981#  5982
        5989#  5996#  5997#  6001#  6014#  6015#  6049#  6079#  6092#  6093#  6128#  6158#  6171#  6172#  6207#
        6257#  6270#  6271#  6307#  6357#  6369#  6370#  6394#  6398#  6425#  6426#  6427#  6428#  6435#  6452#
        6457#  6458   6459   6460   6463#  6464   6465   6466   6467   6469#  6495#  6500#  6501   6502   6505#
        6506   6507   6508   6509   6512#  6513   6514   6519#  6559#  6560#  6561#
M$GNLS     1#  1014#
M$GNSU     1#  1014#
M$GNTA     1#  1014#  1171#  1192#  2391#  2470#  2502#  2568#  2621#  2669#  2722#  2767#  2816#  2893#  4474#
```

```
                4688#  4707#  4734#  4773#  4795#  4902#  4982#  5069#  5138#  5206#  5285#  5362#  5435#  5509#  5588#
                5671#  5767#  5864#  6000#  6078#  6157#  6256#  6356#  6397#  6434#  6469#  6470   6519#  6520
M$GNTE             1#  1014#  4813#  4912#  4992#  5080#  5149#  5217#  5296#  5373#  5450#  5522#  5600#  5682#  5780#
                5878#  6012#  6090#  6169#  6268#  6367#  6410#
M$HAPT             1#  1014#  1034#
M$HNAP             1#  1014#  1034#  1073
M$INCR             1#  1014#  1024#  1163#  1184#  2362#  2372#  2380#  2386#  2392#  2443#  2456#  2463#  2471#  2491#
                2498#  2503#  2528#  2535#  2564#  2569#  2594#  2604#  2611#  2618#  2622#  2645#  2654#  2663#  2670#
                2693#  2700#  2714#  2723#  2746#  2755#  2763#  2768#  2789#  2804#  2812#  2817#  2839#  2848#  2874#
                2886#  2894#  3306#  3315#  3318#  3489#  3646#  3717#  3815#  3854#  3868#  3932#  3950#  4115#  4423#
                4465#  4475#  4485#  4508#  4512#  4518#  4524#  4530#  4536#  4542#  4558#  4565#  4609#  4629#  4675#
                4685#  4689#  4703#  4708#  4719#  4724#  4727#  4735#  4746#  4752#  4774#  4786#  4796#  4813#  4814#
                4883#  4890#  4893#  4903#  4912#  4913#  4918#  4936#  4961#  4968#  4975#  4979#  4983#  4992#  4993#
                4998#  5016#  5048#  5055#  5062#  5066#  5070#  5080#  5081#  5083#  5124#  5131#  5135#  5139#  5149#
                5150#  5152#  5192#  5199#  5203#  5207#  5217#  5218#  5220#  5264#  5271#  5278#  5282#  5286#  5296#
                5297#  5299#  5341#  5348#  5355#  5359#  5363#  5373#  5374#  5376#  5420#  5428#  5432#  5436#  5450#
                5451#  5453#  5495#  5506#  5510#  5522#  5523#  5525#  5570#  5577#  5585#  5589#  5600#  5601#  5603#
                5660#  5668#  5672#  5682#  5683#  5685#  5727#  5743#  5750#  5756#  5764#  5768#  5780#  5781#  5783#
                5854#  5861#  5865#  5878#  5879#  5881#  5981#  5989#  5997#  6001#  6012#  6013#  6015#  6049#  6079#
                6090#  6091#  6093#  6128#  6158#  6169#  6170#  6172#  6207#  6257#  6268#  6269#  6271#  6307#  6357#
                6367#  6368#  6370#  6394#  6398#  6410#  6411#  6425#  6435#  6452#  6495#
M$IOSE             1#  1014#
M$LDRO             1#  1014#  4511#  4517#  4523#  4529#  4541#  4564#  4628#  4684#  4889#  4917#  4978#  4997#  5065#
                5082#  5134#  5151#  5202#  5219#  5281#  5298#  5358#  5375#  5431#  5452#  5505#  5524#  5584#  5602#
                5667#  5684#  5763#  5782#  5860#  5880#  5996#  6014#  6092#  6171#  6270#  6369#
M$MASK             1#  1014#
M$MCHI             1#  1014#
M$MCLO             1#  1014#
M$MSK1             1#  1014#
M$POP              1#  1014#  1171#  1192#  2391#  2470#  2502#  2568#  2621#  2669#  2722#  2767#  2816#  2893#  4474#
             4492#  4688#  4707#  4734#  4773#  4795#  4902#  4982#  5069#  5138#  5206#  5285#  5362#  5435#  5509#
             5588#  5671#  5767#  5864#  6000#  6078#  6157#  6256#  6356#  6397#  6434#  6469#  6519#  6564#
M$PRIN             1#  1014#  2369#  2377#  2383#  2452#  2459#  2494#  2531#  2559#  2599#  2607#  2614#  2650#  2659#
             2696#  2709#  2750#  2758#  2799#  2807#  2844#  2871#  2880#  3312#  4671#  4748#  5976#
M$PUSH             1#  1014#  1024#  1163#  1184#  2362#  2443#  2491#  2528#  2594#  2645#  2693#  2746#  2789#  2839#
             4465#  4485#  4508#  4703#  4719#  4746#  4786#  4813#  4814   4912#  4913   4992#  4993   5080#  5081
             5149#  5150   5217#  5218   5296#  5297   5373#  5374   5450#  5451   5522#  5523   5600#  5601   5682#
             5683   5780#  5781   5878#  5879   6012#  6013   6090#  6091   6169#  6170   6268#  6269   6367#  6368
             6410#  6411   6452#  6495#
M$PUT              1#  1014#  2369#  2377#  2383#  2452#  2459#  2494#  2531#  2559#  2599#  2607#  2614#  2650#  2659#
             2696#  2709#  2750#  2758#  2799#  2807#  2844#  2871#  2880#  3312#  4554#  4671#  4748#  5976#
M$PUT1             1#  1014#  2369#  2370   2377#  2378   2383#  2384   2452#  2453   2454   2459#  2460   2461   2494#
             2495   2496   2531#  2532   2533   2559#  2560   2561   2562   2599#  2600   2601   2602   2607#  2608
             2609   2614#  2615   2616   2650#  2651   2652   2659#  2660   2661   2696#  2697   2698   2709#  2710
             2711   2712   2750#  2751   2752   2753   2758#  2759   2760   2761   2799#  2800   2801   2802   2807#
             2808   2809   2810   2844#  2845   2846   2871#  2872   2880#  2881   2882   2883   2884   3312#  3313
             4554#  4555   4556   4557   4671#  4672   4673   4748#  4749   4750   5976#  5977   5978   5979
M$RADI             1#  1014#  6457#  6463#  6500#  6505#  6512#
M$RBRO             1#  1014#
M$RNRO             1#  1014#  4541#  4543   4628#  4630
M$SETS             1#  1014#  1024#  1163#  1184#  2362#  2443#  2491#  2528#  2594#  2645#  2693#  2746#  2789#  2839#
             4465#  4485#  4508#  4703#  4719#  4746#  4786#  4814#  4913#  4993#  5081#  5150#  5218#  5297#  5374#
             5451#  5523#  5601#  5683#  5781#  5879#  6013#  6091#  6170#  6269#  6368#  6411#  6452#  6495#
M$STAR             1#  1014#
M$SVC              1#  1014#  2369#  2372   2377#  2380   2383#  2386   2391#  2392   2452#  2456   2459#  2463   2470#
             2471   2494#  2498   2502#  2503   2531#  2535   2559#  2564   2568#  2569   2599#  2604   2607#  2611
```

```
          2614#  2618   2621#  2622   2650#  2654   2659#  2663   2669#  2670   2696#  2700   2709#  2714   2722#
          2723   2750#  2755   2758#  2763   2767#  2768   2799#  2804   2807#  2812   2816#  2817   2844#  2848
          2871#  2874   2880#  2886   2893#  2894   3306   3312#  3315   3318#  3489#  3646#  3717#  3815#  3854#
          3868#  3932#  3950#  4115#  4423#  4468#  4474#  4475   4511#  4512   4517#  4518   4523#  4524   4529#
          4530   4536#  4541#  4542   4554#  4558   4564#  4565   4609#  4628#  4629   4671#  4675   4684#  4685
          4688#  4689   4707#  4708   4724#  4727#  4734#  4735   4748#  4752   4767#  4773#  4774   4788#  4795#
          4796   4883   4889#  4890   4893#  4902#  4903   4917#  4918   4936#  4961#  4968#  4975#  4978#  4979
          4982#  4983   4997#  4998   5016#  5048#  5055#  5062#  5065#  5066   5069#  5070   5082#  5083   5124#
          5131#  5134#  5135   5138#  5139   5151#  5152   5192#  5199#  5202#  5203   5206#  5207   5219#  5220
          5264#  5271#  5278#  5281#  5282   5285#  5286   5298#  5299   5341#  5348#  5355#  5358#  5359   5362#
          5363   5375#  5376   5420#  5428#  5431#  5432   5435#  5436   5452#  5453   5495   5505#  5506   5509#
          5510   5524#  5525   5570#  5577#  5584#  5585   5588#  5589   5602#  5603   5660#  5667#  5668   5671#
          5672   5684#  5685   5727#  5743#  5750#  5756#  5763#  5764   5767#  5768   5782#  5783   5854#  5860#
          5861   5864#  5865   5880#  5881   5976#  5981   5989#  5996#  5997   6000#  6001   6014#  6015   6049#
          6078#  6079   6092#  6093   6128#  6157#  6158   6171#  6172   6207#  6256#  6257   6270#  6271   6307#
          6356#  6357   6369#  6370   6394#  6397#  6398   6425   6434#  6435
M$TLAB       1#  1014#  2372#  2380#  2386#  2392#  2456#  2463#  2471#  2498#  2503#  2535#  2564#  2569#  2604#
          2611#  2618#  2622#  2654#  2663#  2670#  2700#  2714#  2723#  2755#  2763#  2768#  2804#  2812#  2817#
          2848#  2874#  2886#  2894#  3306#  3315#  3318#  3489#  3646#  3717#  3815#  3854#  3868#  3932#  3950#
          4115#  4423#  4475#  4512#  4518#  4524#  4530#  4536#  4542#  4558#  4565#  4609#  4629#  4675#  4685#
          4689#  4708#  4724#  4727#  4735#  4752#  4774#  4796#  4883#  4890#  4893#  4903#  4918#  4936#  4961#
          4968#  4975#  4979#  4983#  4998#  5016#  5048#  5055#  5062#  5066#  5070#  5083#  5124#  5131#  5135#
          5139#  5152#  5192#  5199#  5203#  5207#  5220#  5264#  5271#  5278#  5282#  5286#  5299#  5341#  5348#
          5355#  5359#  5363#  5376#  5420#  5428#  5432#  5436#  5453#  5495#  5506#  5510#  5525#  5570#  5577#
          5585#  5589#  5603#  5660#  5668#  5672#  5685#  5727#  5743#  5750#  5756#  5764#  5768#  5783#  5854#
          5861#  5865#  5881#  5981#  5989#  5997#  6001#  6015#  6049#  6079#  6093#  6128#  6158#  6172#  6207#
          6257#  6271#  6307#  6357#  6370#  6394#  6398#  6425#  6435#
M$TSTL       1#  1014#  2372#  2380#  2386#  2392#  2456#  2463#  2471#  2498#  2503#  2535#  2564#  2569#  2604#
          2611#  2618#  2622#  2654#  2663#  2670#  2700#  2714#  2723#  2755#  2763#  2768#  2804#  2812#  2817#
          2848#  2874#  2886#  2894#  3306#  3315#  3318#  3489#  3646#  3717#  3815#  3854#  3868#  3932#  3950#
          4115#  4423#  4475#  4512#  4518#  4524#  4530#  4536#  4542#  4558#  4565#  4609#  4629#  4675#  4685#
          4689#  4708#  4724#  4727#  4735#  4752#  4774#  4796#  4883#  4890#  4893#  4903#  4918#  4936#  4961#
          4968#  4975#  4979#  4983#  4998#  5016#  5048#  5055#  5062#  5066#  5070#  5083#  5124#  5131#  5135#
          5139#  5152#  5192#  5199#  5203#  5207#  5220#  5264#  5271#  5278#  5282#  5286#  5299#  5341#  5348#
          5355#  5359#  5363#  5376#  5420#  5428#  5432#  5436#  5453#  5495#  5506#  5510#  5525#  5570#  5577#
          5585#  5589#  5603#  5660#  5668#  5672#  5685#  5727#  5743#  5750#  5756#  5764#  5768#  5783#  5854#
          5861#  5865#  5881#  5981#  5989#  5997#  6001#  6015#  6049#  6079#  6093#  6128#  6158#  6172#  6207#
          6257#  6271#  6307#  6357#  6370#  6394#  6398#  6425#  6435#
M$WORD       1#  1014#  1073#  1082   1128#  1130   1131   1132   1133   1134   1135   1136   1137   1138   1139
          1140   1141   1142   1143   1144   1145   1146   1147   1148   1149   3306#  3307   3308   3309   4468#
          4727#  4767#  4788#  4883#  4884   4885   4886   5495#  5496   5497   5498   6425#  6426   6427   6428
          6457#  6463#  6500#  6505#  6512#  6560   6561
M$XFER       1#  1014#
OPEN         1#  1014#
PASS      1544#  2388   2466   2666   2889   3007   3044   3069   3113   3143   3227   3280   3320   3392   3504
          3604   3648   3727   3760   3953   3991   4034   4127   4223   4271   4386
POINTE       1#  1014#  1030
PRINTB       1#  1014#  2368   2376   2451   2458   2493   2530   2598   2649   2658   2695   2749   2798   2843
          5975
PRINTF       1#  1014#  3311   4670   4747
PRINTS       1#  1014#
PRINTX       1#  1014#  2382   2558   2606   2613   2708   2757   2806   2870   2879
READBU       1#  1014#
READEF       1#  1014#  4510   4516   4522   4528
RFLAGS       1#  1014#
SAVE      1503#  2363   2444   2646   2840   2930   3036   3063   3095   3134   3186   3271   3302   3364   3436
```

```
CVDHAA0 DHV-11 FUNC TST PART1   MACY11 30A(1052)  12-JUL-83  10:52  PAGE 177
CVDHAA.P11    12-JUL-83 00:42          CROSS REFERENCE TABLE -- MACRO NAMES

          3535    3633    3676    3751    3794    3979    4014    4110    4153    4256    4317
SETPRI      1#   1014#   4563    4683    4916    4977    4996    5064    5081    5133    5150    5201    5218    5280    5297
          5357    5374    5430    5451    5504    5523    5583    5601    5666    5683    5762    5781    5859    5879    5995
          6013    6091    6170    6269    6368
SETVEC      1#   1014#   4553
SLASH       1#   1014#
STARS       1#   1014#
SVC         1#   1012#   1013
XFER        1#   1014#   4468#   4727#   4767#   4788#
XFERF       1#   1014#
XFERT       1#   1014#


. ABS.  026220     000


 ERRORS DETECTED:  0

 CVDHAA.BIC,CVDHAA.LST/CRF/NL:TOC/SOL=SVC34R.MLB,CVDHAA.P11
 RUN-TIME: 18 25 2 SECONDS
 RUN-TIME RATIO: 335/46=7.1
 CORE USED:  16K  (31 PAGES)
```