DHU11

DHU-11 FUNC TST PART 1

CZDHUA0

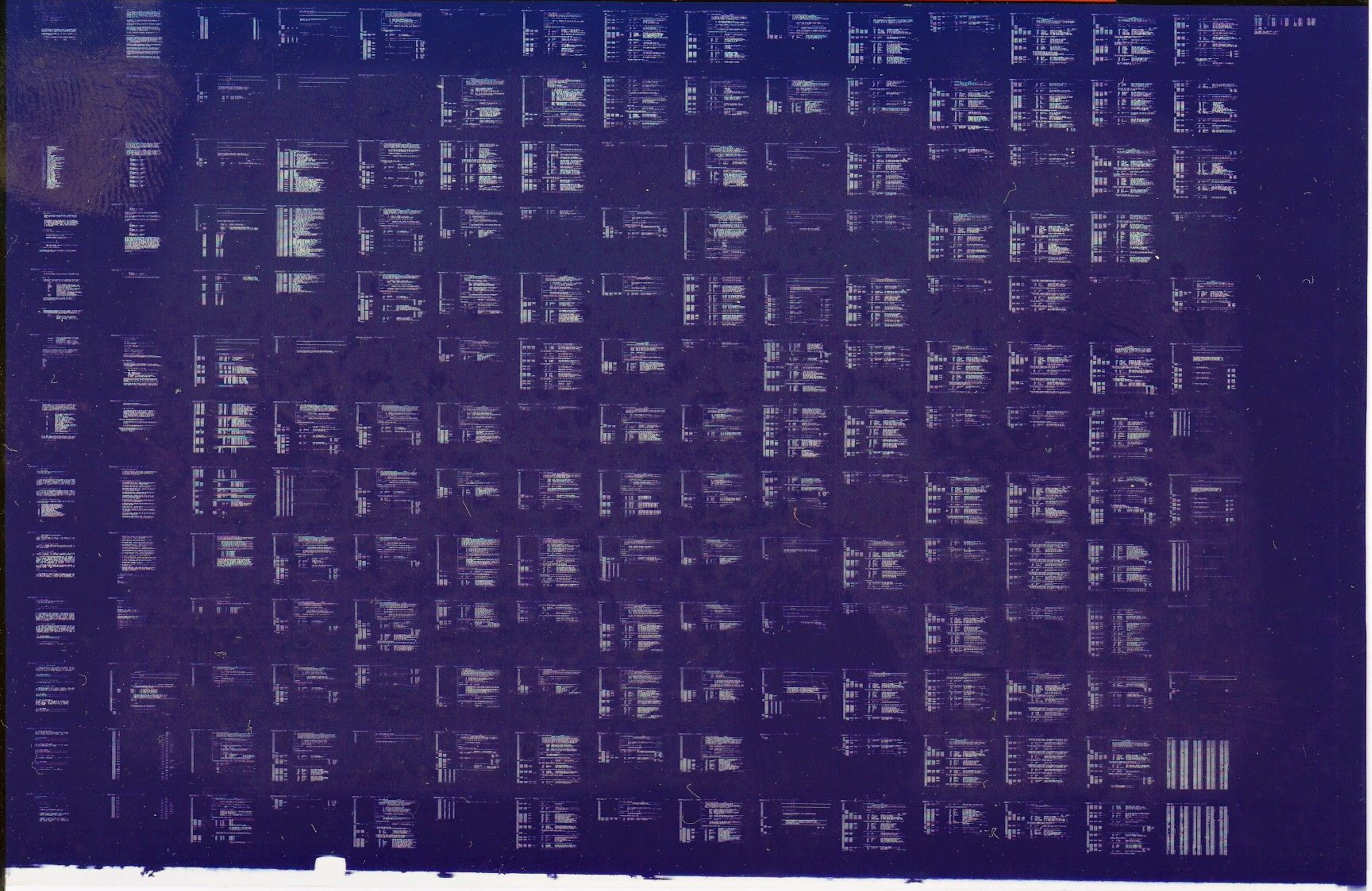
AH-T795A-MC

FICHE 1 OF 1

APR 1984

digital

Made In USA



PROGRAM DOCUMENT

```
.REM &

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

PRODUCT CODE:       AC-T794A-MC

PRODUCT NAME:       CZDHUAO DHU-11 FUNC TST PART1

PRODUCT DATE:       15 DECEMBER 1983

MAINTAINER:         ENE - DIAGNOSTICS GROUP

AUTHOR:             ANTHONY HART

MODIFIED BY:
```

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

ORIGINAL RELEASE:       15-DEC-83       ANTHONY HART
```

TABLE OF CONTENTS

1.0 GENERAL PROGRAM CONSIDERATIONS

1.1 PROGRAM ABSTRACT

CZDHUAO IS PART ONE OF THE DHU FUNCTIONAL VERIFICATION TEST. THIS PART OF THE TEST VERIFIES THE RESET, SELFTEST, REGISTER ACCESS, AND INTERRUPT FUNCTIONS OF THE BOARD ARE FUNCTIONING CORRECTLY.

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

1.2 SYSTEM REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DHU FVT:

- O UNIBUS PROCESSOR WITH AT LEAST 32K BYTES OF MEMORY.
- O DHU BOARDS INSTALLED ON THE UNIBUS.
- O APPROPRIATE PROGRAM LOAD DEVICE SUPPORTING XXDP+ MEDIA OR A DOWN LINE LOADING SYSTEM.

1.3 RELATED DOCUMENTS AND STANDARDS

- O XXDP+ USER'S MANUAL - DESCRIBES THE RUNNING OF DIAGNOSTICS UNDER THE XXDP+ MONITOR.

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE PROCESSOR, THE UNIBUS, 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 UNITS SWITCH SHOULD NOT BE USED WITH THIS PROGRAM. 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 -
THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

2.4.10 DISPLAY COMMAND -
****
DIS(PLAY)/UNITS:<UNIT-LIST>
****

2.4.10.1 EFFECT OF DISPLAY COMMAND -

THE HARDWARE P-TABLE FOR THE TEST STATION IS PRINTED IN THE FORMAT IN WHICH IT WAS 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 REQUESTES THE CSR ADDRESS OF THE SPECIFIED DHU-11. THE DEFAULT ANSWER FOR THIS QUESTION IS ADDRESS 160460 (OCTAL).

2. INTERRUPT VECTOR ADDRESS - THIS QUESTION REQUESTS THE INTERRUPT VECTOR ADDRESS OF THE SPECIFIED DHU-11. THE DEFAULT ANSWER IS 310 (OCTAL).

3. ACTIVE LINES BIT MAP - THIS QUESTION REQUESTS AN OCTAL BIT MAP OF THE SERIAL COMMUNICATION LINES ON THE DHU11 WHICH ARE BEING SELECTED FOR TESTING. IF THE BIT IN THE BIT MAP IS SET WHICH CORRESPONDS TO A PARTICULAR LINE (I.E. BIT 5 FOR LINE 5) THAT LINE WILL BE TESTED BY THE FVT.

4. BR LEVEL - THIS QUESTION REQUESTS THE INTERRUPT BR LEVEL OF THE SPECIFIED DHU-11. THE DEFAULT ANSWER IS BR 5.

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 SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

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 THAT UNIT.

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

3. EXTENDED ERROR REPORTING - THIS QUESTION ASKS WHETHER EXTENDED ERROR INFORMATION IS REQUIRED OTHER THAN THE "TEST FAILED" MESSAGE, ON EACH ERROR REPORTED. THE DEFAULT IS "NO" I.E. ONLY A MESSAGE REPORTING THE FACT THAT THE TEST FAILED WILL BE PRINTED.

4. NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE - THIS QUESTION IS ASKED ONLY IF THE PREVIOUS QUESTION WAS ANSWERED "YES". THE QUESTION ASKS FOR THE NUMBER OF DATA ERRORS WHICH

SHOULD BE REPORTED INDIVIDUALLY BY THIS PROGRAM FOR EACH LINE
FOR EACH TRANSMISSION TEST. ERRORS WHICH ARE NOT REPORTED
INDIVIDUALLY ARE REPORTED IN SUMMARY ERROR REPORTS.

2.7 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

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

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

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

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

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

```
        UNIT 8
        CSR ADDRESS (O)  160000<CR>
        SUB-DEVICE # (O) ?  7<CR>
        Q-FACTOR (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 (0) ?  160000<CR>
SUB-DEVICE # (0) ?  0-7<CR>
Q-FACTOR (0) 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) 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 SPECIFIC ERROR MESSAGES

THIS PROGRAM IS INTENDED TO PROVIDE A GO/NOGO INDICATION
OF THE FUNCTIONALITY OF THE DHU-11 BOARDS. TO EXECUTE THE
PROGRAM IN THIS MODE THE OPERATOR NEED ONLY ANSWER THE
"EXTENDED ERROR REPORTING" SOFTWARE QUESTION WITH "NO", THE
PROGRAM WILL THEN ONLY PRINT THE NAME OF THE FAILING TEST
THE TEST AND ERROR NUMBERS. FOR A LIST OF THE TEST NAMES
IN THIS PROGRAM SEE THE TEST SUMMARIES SECTION OF THIS
DOCUMENT. AN EXAMPLE OF SUCH A AN ERROR MESSAGE IS THE
FOLLOWING:

```
CZDHU DVC FTL ERR 01603 ON UNIT 02 TST 15 SUB 000 PC: XXXXXX
DEVICE REGISTER WORD READ/WRITE TEST FAILED.
```

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

IF THE OPERATOR HAD REQUESTED EXTENDED ERROR REPORTING THE
SAME WOULD BE REPORTED AS FOLLOWS:

```
CZDHU DVC FTL ERR 01603 ON UNIT 02 TST 15 SUB 000 PC: XXXXXX
DEVICE REGISTER WORD READ/WRITE TEST FAILED.
  BAD BIT(S) IN DEVICE TBUFFAD1 REGISTER FOR LINE 7 (D).
    EXPECTED DATA: 000000 (0).
    ACTUAL DATA:   000023 (0).
```

4.0 PERFORMANCE AND PROGRESS REPORTS

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

5.0 TEST SUMMARIES

THE FOLLOWING ARE INCLUDED WITHIN CZDHUA:

1. DEVICE REGISTER ACCESS TEST - VERIFIES THAT THE UUT REGISTERS WILL RESPOND WITH THE CORRECT UNIBUS HANDSHAKING SIGNALS. VERIFIES THAT THE UUT IS AT THE CORRECT 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 CLEARS WITHIN A SHORT TIME AFTER IT IS SET WHEN THE SKIP SELFTEST SEQUENCE IS USED.

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

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

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

7. RX.DATA.VALID TEST - VERIFIES THAT THE RX.DATA.VALID BIT IS SET FOR ALL THE CODES IN THE FIFO AND CLEAR AFTER ALL THE CODES HAVE BEEN READ.

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

9. BMP CHECK TEST - VERIFIES THAT THE DUT DOES NOT IMMEDIATELY FAIL THE BACKGROUND MONITOR PROGRAM, AS THIS MAY INVALIDATE FURTHER TESTS.

10. SKIP SELFTEST TEST - VERIFIES THAT THE DUT SKIPS THE SELFTEST IN THE TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS COMPLEATION.

11. DIAGNOSTIC.FAIL (SKIP SELFTEST) TEST - VERIFIES USING THE SKIP SELFTEST SEQUENCE THAT THE DIAG.FAIL BIT GOES TO BOTH THE ACTIVE AND INACTIVE STATES WITHIN THE ALLOWED TIMES.

12. SELFTEST TEST - VERIFIES THAT THE DUT'S SELFTEST EXECUTES WITHIN THE CORRECT TIME AND THAT THE CORRECT CODES ARE RETURNED IN THE FIFO AFTER ITS COMPLEATION.

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

14. ROM VERSION NUMBER - VERIFIES THAT THE ROM VERSION NUMBERS ARE REPORTED CORRECTLY AND IF REQUESTED PRINTS THEM OUT.

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

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

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

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

19. ID.BIT TEST - VERIFIES THAT THE ID BIT READS AS SET.

20. TX.ENABLE (INACTIVE) TEST - VERIFIES THAT WHEN A LINE'S TX.ENBL BIT IS CLEAR, TRANSMISSION WILL NOT TAKE PLACE ON THAT LINE.

21. TX.ENABLE (ACTIVE) TEST - VERIFIES THAT WHEN A LINE'S TX.ENBL BIT IS SET, TRANSMISSION WILL TAKE PLACE ON THAT LINE.

22. INTERRUPT TEST - VERIFIES THAT THE DUT WILL GENERATE RECEPTION AND TRANSMISSION INTERRUPTS CORRECTLY.

23. BR LEVEL TEST - VERIFIES THAT THE DUT INTERRUPTS AT THE CORRECT BUS REQUEST LEVEL.

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

6.0 EXAMPLE ERROR FREE PASS

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

```
.R CZDHUAO
CZDHUAO.BIN


DRS
CZDHU-A-0
DHU-11 FUNC TST PART1
UNIT IS DHU-11
```

K2


```
RESTRT ADDR:   147670
DR>STA/PAS:1

CHANGE HW (L)   ? Y

# UNITS (D)  ?  2

UNIT 0
CSR ADDRESS:  (0)  160460 ? ↑Z

UNIT 1
CSR ADDRESS:  (0)  160460 ?  160500
INTERRUPT VECTOR ADDRESS:  (0)  310  ? 320
ACTIVE LINE BIT MAP:  (0)  177777  ? <CR>
INTERRUPT BR LEVEL:  (0)  5  ? <CR>

CHANGE SW  (L)  ? Y

REPORT UNIT NUMBER AS EACH UNIT IS TESTED:  (L) Y ? <CR>
ROM VERSION PRINTOUT ON THE FIRST PASS:  (L) Y ? <CR>
EXTENDED ERROR REPORTING:  (L) N  ? Y
NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE:  (D)  0 ? 1

TESTING UNIT  :  0(D)

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

TESTING UNIT  :  1(D)

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

CZDHU EOP       1
    0 TOTAL ERRS

DR>
```

```
                        E

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

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


                        .SBTTL  PROGRAM HEADER


                                .MCALL  SVC
000000                          SVC                                     ; INITIALIZE SUPERVISOR MACROS

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

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

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

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

002000                          BGNMOD

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

002000                          POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL


002000                          HEADER  CZDHU,A,0,16,0,PRI07
002000                                                          L$NAME::
002000  103                                                             .ASCII /C/
002001  132                                                             .ASCII /Z/
```

PROGRAM HEADER

```
002002      104                                         .ASCII /D/
002003      110                                         .ASCII /H/
002004      125                                         .ASCII /U/
002005      000                                         .BYTE   0
002006      000                                         .BYTE   0
002007      000                                         .BYTE   0
002010                                          L$REV::
002010      101                                         .ASCII  /A/
002011                                          L$DEPO::
002011      060                                         .ASCII  /0/
002012                                          L$UNIT::
002012  000000                                          .WORD   0
002014                                          L$TIML::
002014  000016                                          .WORD   16
002016                                          L$HPCP::
002016  035722                                          .WORD   L$HARD
002020                                          L$SPCP::
002020  036114                                          .WORD   L$SOFT
002022                                          L$HPTP::
002022  002206                                          .WORD   L$HW
002024                                          L$SPTP::
002024  002220                                          .WORD   L$SW
002026                                          L$LADP::
002026  036476                                          .WORD   L$LAST
002030                                          L$STA::
002030  000000                                          .WORD   0
002032                                          L$CO::
002032  000000                                          .WORD   0
002034                                          L$DTYP::
002034  000000                                          .WORD   0
002036                                          L$APT::
002036  000000                                          .WORD   0
002040                                          L$DTP::
002040  002124                                          .WORD   L$DISPATCH
002042                                          L$PRIO::
002042  000340                                          .WORD   PRIO7
002044                                          L$ENVI::
002044  000000                                          .WORD   0
002046                                          L$EXP1::
002046  000000                                          .WORD   0
002050                                          L$MREV::
002050      003                                         .BYTE   C$REVISION
002051      003                                         .BYTE   C$EDIT
002052                                          L$EF::
002052  000000                                          .WORD   0
002054  000000                                          .WORD   0
002056                                          L$SPC::
002056  000000                                          .WORD   0
002060                                          L$DEVP::
002060  004032                                          .WORD   L$DVTYP
002062                                          L$REPP::
002062  024102                                          .WORD   L$RPT
002064                                          L$EXP4::
002064  000000                                          .WORD   0
002066                                          L$EXP5::
002066  000000                                          .WORD   0
002070                                          L$AUT::
```

PROGRAM HEADER

```
002070  000000                          .WORD   0
002072                          L$DUT::
002072  024756                          .WORD   L$DU
002074                          L$LUN::
002074  000000                          .WORD   0
002076                          L$DESP::
002076  004042                          .WORD   L$DESC
002100                          L$LOAD::
002100  104035                          EMT     E$LOAD
002102                          L$ETP::
002102  003762                          .WORD   L$ERRTBL
002104                          L$ICP::
002104  024116                          .WORD   L$INIT
002106                          L$CCP::
002106  024740                          .WORD   L$CLEAN
002110                          L$ACP::
002110  024736                          .WORD   L$AUTO
002112                          L$PRT::
002112  024110                          .WORD   L$PROT
002114                          L$TEST::
002114  000000                          .WORD   0
002116                          L$DLY::
002116  000000                          .WORD   0
002120                          L$HIME::
002120  000000                          .WORD   0
1112
```

```
1124                                    .SBTTL  DISPATCH TABLE
1125
1126                                    ;++
1127                                    ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1128                                    ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1129                                    ;--
1130
1131 002122                                    DISPATCH 24
     002122  000030                                                          .WORD   24
     002124                                                        L$DISPATCH::
     002124  025074                                                  .WORD   T1
     002126  025356                                                  .WORD   T2
     002130  025606                                                  .WORD   T3
     002132  026052                                                  .WORD   T4
     002134  026250                                                  .WORD   T5
     002136  026442                                                  .WORD   T6
     002140  026660                                                  .WORD   T7
     002142  027066                                                  .WORD   T8
     002144  027274                                                  .WORD   T9
     002146  027470                                                  .WORD   T10
     002150  027704                                                  .WORD   T11
     002152  030132                                                  .WORD   T12
     002154  030426                                                  .WORD   T13
     002156  030714                                                  .WORD   T14
     002160  031314                                                  .WORD   T15
     002162  031562                                                  .WORD   T16
     002164  031752                                                  .WORD   T17
     002166  032270                                                  .WORD   T18
     002170  032532                                                  .WORD   T19
     002172  032646                                                  .WORD   T20
     002174  033164                                                  .WORD   T21
     002176  033540                                                  .WORD   T22
     002200  034676                                                  .WORD   T23
     002202  035640                                                  .WORD   T24
1132
```

```
                                ;*****************************************************************************
                                ;
                                ;               FVTA.DHT
                                ;
                                ;*****************************************************************************


                                .SBTTL  DEFAULT HARDWARE P-TABLE

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

002204                                  BGNHW   DFPTBL
002204  000004                                                                 .WORD   L10000-L$HW/2
002206                                                                  L$HW::
002206                                                                  DFPTBL::

002206  160460                          .WORD   160460  ;DEFAULT CSR ADDRESS
002210  000310                          .WORD   310     ;DEFAULT VECTOR ADDRESS
002212  177777                          .WORD   177777  ;DEFAULT ACTIVE LINES BIT MAP
002214     005                          .BYTE   5       ;DEFAULT BR LEVEL
                                        .EVEN

                                        ENDHW
002216
002216                                                                  L10000:
```

```
    1167
    1168                        ;*****************************************************************************************
*
    1169                        ;
    1170                        ;               FVTA.SWT
    1171                        ;
    1172                        ;*****************************************************************************************
    1173
    1174
    1175
    1176                        .SBTTL  SOFTWARE P-TABLE
    1177
    1178                        ;++
    1179                        ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
    1180                        ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
    1181                        ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
    1182                        ; AT RUN TIME.
    1183                        ;--
    1184
    1185 002216                         BGNSW   SFPTBL
         002216  000002                                                         .WORD   L10001-L$SW/2
         002220                                                         L$SW::
         002220                                                         SFPTBL::
    1186
    1187 002220  000021                 OPTION::        .WORD   21      ;BIT MAP OF PROGRAM CONTROL FLAGS
    1188 002222  000000                 NDERPT::        .WORD   0       ;DEFAULT NUMBER OF INDIVIDUAL DATA ERRORS TO RPT.
    1189
    1190 002224                         ENDSW
         002224                                                         L10001:
```

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


                        .SBTTL  GLOBAL EQUATES SECTION




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

000020                          NUMLNS==20      ;NUMBER OF LINES ON DHU11 IS 16.
177777                          MAPLNS==177777  ;BIT MAP OF LINES ON DHU11.

                        ;****** DEVICE REGISTER OFFSETS FROM THE CSR'S ADDRESS ******
000004                          LPRO==4         ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
000006                          FSLSO==6        ;FIFOSIZE/STATUS REGISTER OFFSET FROM THE CSR ADDRESS
000016                          TXBFCO==16      ;TRANSMIT COUNT REGISTER OFFSET FROM THE CSR ADDRESS

002224                          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
000040                  BIT5==  BIT05
000020                  BIT4==  BIT04
000010                  BIT3==  BIT03
000004                  BIT2==  BIT02
000002                  BIT1==  BIT01
000001                  BIT0==  BIT00
```

```
                        ;
                        ; EVENT FLAG DEFINITIONS
                        ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                        ;
000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
                        ;
                        ;
                        ; PRIORITY LEVEL DEFINITIONS
                        ;
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
000140                  PRI03== 140
000100                  PRI02== 100
000040                  PRI01== 40
000000                  PRI00== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==       4
000010                  LOT==       10
000020                  ADR==       20
000040                  IDU==       40
000100                  ISR==       100
000200                  UAM==       200
000400                  BOE==       400
001000                  PNT==       1000
002000                  PRI==       2000
004000                  IXE==       4000
010000                  IBE==       10000
020000                  IER==       20000
040000                  LOE==       40000
100000                  HOE==       100000

1242
```

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


                        .SBTTL  GLOBAL DATA SECTION

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


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

002224  177777          ACTLNS:: .WORD  177777  ;ACTIVE LINE BIT MAP.
002226  000300          RXVECA:: .WORD  300     ;RX VECTOR ADDRESS.
002230  000304          TXVECA:: .WORD  304     ;TX VECTOR ADDRESS.
002232  000000          UNITN:: .WORD   0       ;UNIT NUMBER.
002234     004          BRLEVL:: .BYTE  4       ;INTERRUPT BUS REQUEST LEVEL
                                 .EVEN


                        ;*****************************************************************************
                        ;       DEVICE REGISTER ADDRESS TABLE
                        ;*****************************************************************************
002236                  DRADRT::
002236  160020          CSRA::  .WORD   160020  ;DMU-11 CSR ADDRESS.
002240  160022  RXTMA:: RBUFA:: .WORD   160022  ;DMU-11 RECIEVE BUFFER/TIMER ADDRESS.
002242  160024          LPRA::  .WORD   160024  ;DMU-11 LINE PARAMETER REGISTER ADDRESS.
002244  160026  FDATA:: FSLSA:: .WORD   160026  ;DMU-11 FIFO SIZE/LINE STATUS REGISTER ADDRESS.
                                                ;AND FIFO DATA REGISTER ADDRESS.
002246  160030          LNCTRA:: .WORD  160030  ;DMU-11 LINE CONTROL REGISTER ADDRESS.
002250  160032          TXAD1A:: .WORD  160032  ;DMU-11 TRANSMIT BUFFER 1 REGISTER ADDRESS
002252  160034          TXAD2A:: .WORD  160034  ;DMU-11 TRANSMIT BUFFER 2 REGISTER ADDRESS
002254  160036          TXBFCA:: .WORD  160036  ;DMU-11 TRANSMIT BUFFER COUNT REGISTER ADDRESS

                        ;*****************************************************************************
                        ;       REGISTER MESSAGE ADDRESS TABLE
                        ;*****************************************************************************
002256  005470          RMATBB:: .WORD  DR00MG  ;ADDRESS OF "CSR" MESSAGE.
002260  005474                   .WORD  DR02MG  ;ADDRESS OF "RBUF" MESSAGE.
002262  005501                   .WORD  DR04MG  ;ADDRESS OF "LPR" MESSAGE.
002264  005505                   .WORD  DR06MG  ;ADDRESS OF "STAT" MESSAGE.
002266  005523                   .WORD  DR10MG  ;ADDRESS OF "LNCTRL" MESSAGE.
002270  005532                   .WORD  DR12MG  ;ADDRESS OF "TBUFFAD1" MESSAGE.
002272  005543                   .WORD  DR14MG  ;ADDRESS OF "TBUFAD2" MESSAGE.
002274  005554                   .WORD  DR16MG  ;ADDRESS OF "TBUFFCT" MESSAGE.

                        ;*****************************************************************************
                        ;       ASSORTED GLOBAL VARIABLES:
                        ;*****************************************************************************
```

```
1301 002276  000000                    BUFPTR:: .WORD  0       ;STORAGE FOR RECEIVE CHARACTER BUFFER POINTER.
1302 002300  000000                    EXOERR:: .WORD  0       ;"EXIT ON ERROR" FLAG.
1303 002302  000000                    CTRLCF:: .WORD  0       ;STORAGE FOR THE CONTROL-C FLAG.
1304 002304  000000                    IESTAT:: .WORD  0       ;STORAGE FOR THE INTERRUPT ENABLE BIT STATES.
1305 002306  000000                    PASCNT:: .WORD  0       ;STO'G FOR PASS COUNT USED IN ROM VERSION TEST.
1306 002310  000000                    RXINTC:: .WORD  0       ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
1307 002312  000000                    RXINTF:: .WORD  0       ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
1308 002314  000000                    TP4FLG:: .WORD  0       ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
1309 002316  000000                    TP4VEC:: .WORD  0       ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
1310 002320  000001                    TSTNUM:: .WORD  1       ;STORAGE FOR THE TEST NUMBER.
1311 002322  000000                    TXINTC:: .WORD  0       ;STORAGE FOR TRANSMIT INTERRUPT COUNT.
1312 002324  000000                    TXINTF:: .WORD  0       ;STORAGE FOR TRANSMIT INTERRUPT FLAGS.
1313 002326  000000                    WORD1::  .WORD  0       ;LOCATION FOR PASSING INDIRECT PARAMETERS.
1314
1315                                ;*****************************************************************************
1316                                ;       LINE TIME CLOCK VARIABLES AND STORAGE.
1317                                ;*****************************************************************************
1318 002330  177546                        CLKCSR:: .WORD  177546  ;CSR ADDRESS OF THE LTC.
1319 002332  000300                        CLKBRL:: .WORD  PRI06   ;INTERRUPT PRIORITY LEVEL OF THE LTC.
1320 002334  000100                        CLKVEC:: .WORD  100     ;INTERRUPT VECTOR ADDRESS OF THE LTC.
1321 002336  000074                        CLKHRZ:: .WORD  60.     ;INTERRUPT FREQUENCY OF THE LTC.
1322 002340  000000                        TIMER1:: .WORD  0       ;HARDWARE CLOCK COUNTER #1.
1323 002342  000000                        TIMER2:: .WORD  0       ;HARDWARE CLOCK COUNTER #2.
1324 002344  000170                        TIMER3:: .WORD  120.    ;HARDWARE BREAK COUNTER LOCATION.
1325 002346  000170                        BCOUNT:: .WORD  120.    ;BREAK COUNT VALUE IN CLOCK TICKS.
1326 002350  000021                        MSTICK:: .WORD  17.     ;NUMBER OF MILLI-SECONDS PER LTC TICK.
1327 002352  000062                        MSLCNT:: .WORD  62      ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.
1328
1329                                ;*****************************************************************************
1330                                ;       MEMORY MANAGEMENT VARIABLES AND FLAGS.
1331                                ;*****************************************************************************
1332 002354  177572                        MMSR0:: .WORD   177572  ;ADDRESS OF MEM MGT STATUS REGISTER #0.
1333 002356  000000                        MMPRES:: .WORD  0       ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
1334 002360  000000                        MMENAB:: .WORD  0       ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).
1335 002362  172340                        PAR0A:: .WORD   172340  ;ADDRESS OF MEM MGT PAR #0.
1336
1337                                ;*****************************************************************************
1338                                ;       BIT MASK TABLE OF UN-USED DHU DEVICE REGISTER BITS.
1339                                ;*****************************************************************************
1340 002364  137660                        UNBTTB::.WORD   137660  ;UNUSED BIT MASK FOR THE CSR
1341 002366  177777                                .WORD   177777  ;UNUSED BIT MASK FOR THE RBUF/RXTIMER REG
1342 002370  000007                                .WORD   7       ;UNUSED BIT MASK FOR THE LPR
1343 002372  177777                                .WORD   177777  ;UNUSED BIT MASK FOR THE STAT/FIFOSIZE/DATA REG
1344 002374  166051                                .WORD   166051  ;UNUSED BIT MASK FOR THE LNCTRL
1345 002376  000000                                .WORD   0       ;UNUSED BIT MASK FOR THE TBUFFAD1
1346 002400  177774                                .WORD   177774  ;UNUSED BIT MASK FOR THE TBUFFAD2
1347 002402  000000                                .WORD   0       ;UNUSED BIT MASK FOR THE TBUFFCT
1348
1349                                ;*****************************************************************************
1350                                ;       TABLE OF WORDS WITH CORRESPONDING BIT SET FOR GENERATION OF BIT MAPS.
1351                                ;*****************************************************************************
1352 002404  000001                BITTBL:: .WORD  1               ;BIT 0 SET.
1353 002406  000002                        .WORD   2               ;BIT 1 SET.
1354 002410  000004                        .WORD   4               ;BIT 2 SET.
1355 002412  000010                        .WORD   10              ;BIT 3 SET.
1356 002414  000020                        .WORD   20              ;BIT 4 SET.
1357 002416  000040                        .WORD   40              ;BIT 5 SET.
```

```
1358 002420  000100                    .WORD   100             ;BIT 6 SET.
1359 002422  000200                    .WORD   200             ;BIT 7 SET.
1360 002424  000400                    .WORD   400             ;BIT 8 SET.
1361 002426  001000                    .WORD   1000            ;BIT 9 SET.
1362 002430  002000                    .WORD   2000            ;BIT 10 SET.
1363 002432  004000                    .WORD   4000            ;BIT 11 SET.
1364 002434  010000                    .WORD   10000           ;BIT 12 SET.
1365 002436  020000                    .WORD   20000           ;BIT 13 SET.
1366 002440  040000                    .WORD   40000           ;BIT 14 SET.
1367 002442  100000                    .WORD   100000          ;BIT 15 SET.
1368
1369                            ;*****************************************************************************
1370                            ;*      GPR SAVE AREA ZERO.
1371                            ;*****************************************************************************
1372 002444                     GPRS0B::                        ;BASE OF GPR SAVE AREA NUMBER ZERO.
1373 002444  000000                    .WORD   0               ;WORD 1, STORAGE FOR R1.
1374 002446  000000                    .WORD   0               ;WORD 2, STORAGE FOR R2.
1375 002450  000000                    .WORD   0               ;WORD 3, STORAGE FOR R3.
1376 002452  000000                    .WORD   0               ;WORD 4, STORAGE FOR R4.
1377 002454  000000                    .WORD   0               ;WORD 5, STORAGE FOR R5.
1378
1379                            ;*****************************************************************************
1380                            ;*      TRANSMISSION AND RECEPTION VARIABLES, POINTERS, AND FLAGS.
1381                            ;*****************************************************************************
1382 002456  000000                    ERSMRF:: .WORD  0       ;ERROR SUMMARY REPORT FLAGS.
1383 002460                             ERCNTB:: .BLKW  16.     ;TABLE OF ERROR COUNTERS.
1384
1385                            ;*****************************************************************************
1386                            ;       STORAGE AREA FOR THE BMP CODE QUEUE.
1387                            ;*****************************************************************************
1388 002520  000000                    BMPCQP:: .WORD  0       ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
1389 002522                             BMPCQB:: .BLKW  64.     ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
1390 002722                             BMPCQE::                ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1391
1392                            ;*****************************************************************************
1393                            ;       GENERAL TABLE AND BUFFER AREA--513 WORDS.
1394                            ;*****************************************************************************
1395 002722                     BUFBAS::                        ;BASE OF MEMORY BUFFER.
1396 002722                     ERLTBL::        .BLKW   128.    ;FIRST HALF OF GENERAL TABLE OR BUFFER.
1397 003322                     BUFMID::        .BLKW   64.     ;SECOND HALF OF GENERAL TABLE OR BUFFER.
1398 003522                     BUF3QT::        .BLKW   64.     ;LAST QUARTER OF THE BUFFER AREA.
1399 003722                     BUFEND::                        ;END OF GENERAL PURPOSE MEMORY BUFFER.
1400 003722                     ENDETB::        .BLKW   16.     ;BUFFER OVERFLOW SPACE.
1401
1402
1403
1416 003762                             ERRTBL
     003762                                                                 L$ERRTBL::
     003762  000000             ERRTYP::        .WORD   0
     003764  000000             ERRNBR::        .WORD   0
     003766  000000             ERRMSG::        .WORD   0
     003770  000000             ERRBLK::        .WORD   0
1417
1418                                    .EVEN
```

GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.

```
                        .SBTTL  GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.
                        ;*************************************************************************
                        ;*      THERE ARE 4 ROUTINES AND MACRO DEFINITIONS USED FOR THE HANDLING OF
                        ;*      GPR VALUES DURING SUBROUTINE CALLS WITHIN THIS PROGRAM.  THE FOUR
                        ;*      ROUTINES/MACRO CALLS HAVE THE FOLLOWING NAMES:
                        ;*
                        ;*      SAVE - MACRO DEFINITION USED AT THE BEGINNING OF A SUBROUTINE TO
                        ;*             SAVE THE GPR CONTENTS FOR LATER RESTORATION.
                        ;*      PASS - MACRO DEFINITION USED AT THE END OF A SUBROUTINE TO RESTORE
                        ;*             THE PREVIOUSLY SAVED GPR CONTENTS AND TO LEAVE THE CONTENTS
                        ;*             OF THE SPECIFIED GPR(S) INTACT (NOT RESTORED).
                        ;*      PREG05 - SUBROUTINE WHICH IS CALLED FROM THE SAVE AND PASS MACRO
                        ;*             EXPANSIONS WHICH ACTUALLY PERFORMS THE ACTIONS ON THE GPRS.
                        ;*
                        ;*      DURING A SUBROUTINE WHICH USES THESE GPR SAVE ROUTINES THE VALUES
                        ;*      OF THE GPRS ARE STORED ON THE STACK IN THE FOLLOWING STACK FRAME:
                        ;*
                        ;*             SP     -> RET PC INTO PREG05 ROUTINE.
                        ;*             SP+2   -> GPR R0 CONTENTS.
                        ;*             SP+4   -> GPR R1 CONTENTS.
                        ;*             SP+6   -> GPR R2 CONTENTS.
                        ;*             SP+8   -> GPR R3 CONTENTS.
                        ;*             SP+10  -> GPR R4 CONTENTS.
                        ;*             SP+12  -> GPR R5 CONTENTS.
                        ;*             SP+14  -> RET PC INTO CALLER OF SUB'TNE WHICH CALLED PREG05.
                        ;*
                        ;*      EACH LEVEL OF SUB'TNE CALLING USES 8 WORDS OF STACK OVERHEAD.
                        ;*      THE SAVE AND PASS MACROS CAN ALSO BE USED IN "STRAIGHT LINE CODE"
                        ;*      TO SAVE AND RESTORE THE GPR VALUES.  IN ANY CASE, AFTER THE
                        ;*      ISSUING OF A PASS CALL THE GPRS WILL BE RESTORED TO THE VALUES
                        ;*      THEY HAD PRIOR TO THE LAST SAVE CALL (EXCEPT FOR THE EXCEPTED,
                        ;*      OR PASSED INTACT, GPRS SPECIFIED AS PARAMETERS TO THE PASS CALL)
                        ;*      AND THE SP WILL ALSO BE RESTORED TO ITS CONDITION BEFORE THE LAST
                        ;*      SAVE CALL.  THE PROGRAMMER MUST BE SURE THAT THE SP HAS THE SAME
                        ;*      VALUE WHEN THE PASS MACRO IS CALLED AS IT HAD IMMEDIATELY AFTER
                        ;*      THE SAVE MACRO WAS CALLED.
                        ;*************************************************************************
```

GPR FRAME ACCESS EQUATES

```
1458                                .SBTTL GPR FRAME ACCESS EQUATES
1459                                        ;+++
1460                                        ;EQUATES THAT ALLOW ACCESS TO THE STACK FRAME.  THESE ARE THE
1461                                        ;OFFSETS INTO THE STACK FOR REGISTERS SAVED DURING THE PREGO5
1462                                        ;ROUTINE.
1463                                        ;---
1464
1465        000036                          LPCSLT==        36      ;OFFSET FOR LAST RETURN PC.
1466        000016                          PCSLOT==        16      ;OFFSET FOR RETURN PC.
1467        000014                          R5SLOT==        14      ;OFFSET FOR R5.
1468        000012                          R4SLOT==        12      ;OFFSET FOR R4.
1469        000010                          R3SLOT==        10      ;OFFSET FOR R3.
1470        000006                          R2SLOT==        6       ;OFFSET FOR R2.
1471        000004                          R1SLOT==        4       ;OFFSET FOR R1.
1472        000002                          ROSLOT==        2       ;OFFSET FOR R0.
```

```
1474                            .SBTTL  GLOBAL MACRO DEFINITION          - SAVE -
1475                            ;*****************************************************************************
1476                            ;*      THIS MACRO IS USED AT THE BEGINNING OF A SUBROUTINE TO SAVE THE
1477                            ;*      CONTENTS OF THE GPRS R0 THRU R5.
1478                            ;*
1479                            ;* INPUTS:      SP - UNCHANGED SINCE SUBROUTINE WAS ENTERED
1480                            ;*              R5SLOT - OFFSET TO STACK SLOT FOR R5 (EQUATED TO 14 OCTAL)
1481                            ;*
1482                            ;* OUTPUTS:     GPR SAVE AREA ON THE STACK IS LOADED WITH THE CONTENTS OF GPRS
1483                            ;*              TOP OF STACK - LOADED WITH THE RETURN ADDRESS INTO PREG05
1484                            ;*
1485                            ;* CALLING SEQUENCE:    SAVE
1486                            ;*
1487                            ;* COMMENTS:    NO ARGUMENTS ARE ALLOWED.
1488                            ;*              THE PASS MACRO SHOULD BE CALLED TO RESTORE THE GPR VALUES.
1489                            ;*
1490                            ;* SUBORDINATE ROUTINES CALLED: PREG05.
1491                            ;*****************************************************************************
1492
1493                                    .MACRO  SAVE
1494                                    .LIST
1495                                            JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
1496                                    .NLIST
1497                                    .ENDM   SAVE
```

```
1499                    .SBTTL  GLOBAL MACRO DEFINITION          - PASS -
1500                    ;*****************************************************************************
1501                    ;*      THIS MACRO IS USED IN CONJUNCTION WITH THE SAVE MACRO.  IT IS
1502                    ;*      CALLED AT END OF A SUBROUTINE TO PASS PARAMETERS IN GPRS BACK TO THE
1503                    ;*      CALLING ROUTINE BY ALTERING THE GPR SAVE AREA ON THE STACK AND THEN
1504                    ;*      RETURNING TO PREGO5 TO RESTORE THE GPRS TO THEIR SAVED VALUES.
1505                    ;*
1506                    ;* INPUTS:      ONLY ALLOWED ARGUMENTS ARE "R0" THRU "R5".
1507                    ;*              ROSLOT THRU R5SLOT MUST BE EQUATED TO THEIR RESPECTIVE GPR SAVE
1508                    ;*                      SLOT OFFSETS BEFORE CALLING THIS MACRO.
1509                    ;*
1510                    ;* OUTPUTS:     THE GPR VALUES ARE PUT IN THEIR RESPECTIVE SLOTS ON THE STACK.
1511                    ;*
1512                    ;* CALLING SEQUENCE:    PASS    R0,R1,...
1513                    ;*
1514                    ;* COMMENTS:    ANY COMBINATION OF GPR ARGUMENTS MAY BE LISTED IN ANY ORDER.
1515                    ;*              FOR EXAMPLE, THE FOLLOWING ARE LEGAL:
1516                    ;*                      PASS    R1
1517                    ;*                      PASS    R4,R0,R2
1518                    ;*              THE GPRS LISTED AS ARGUMENTS WILL BE PASSED INTACT TO THE
1519                    ;*              CALLING ROUTINE, ALL OTHER GPRS WILL BE RESTORED.
1520                    ;*              THE SP MUST BE AT ITS ORIGINAL VALUE WHEN PASS IS CALLED.
1521                    ;*
1522                    ;*              THE MACRO CALL
1523                    ;*                      PASS    R0,R3
1524                    ;*              EXPANDS INTO THE FOLLOWING ASSEMBLY CODE:
1525                    ;*                      MOV     R0,ROSLOT(SP)           ;PUT R0 IN STACK SLOT.
1526                    ;*                      MOV     R3,R3SLOT(SP)           ;PUT R3 IN STACK SLOT.
1527                    ;*                      JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
1528                    ;*              IN THIS EXAMPLE GPRS R1, R2, R4, AND R5 WILL BE RESTORED TO
1529                    ;*              THEIR VALUES CONTAINED IN THE STACK FRAME AND R0 AND R3
1530                    ;*              WILL BE LEFT AT THEIR VALUES PRIOR TO THIS PASS CALL.
1531                    ;*
1532                    ;* SUBORDINATE ROUTINES CALLED: (PREGRT - LABEL WITHIN PREGO5, VALUE ON STACK.)
1533                    ;*****************************************************************************
1534
1535                            .MACRO  PASS    A,B,C,D,E,F
1536                            .IRP    X,<A,B,C,D,E,F>
1537                            .IF     NB,X
1538                            .LIST
1539                                    MOV     X,X'SLOT(SP)            ;PUT X IN STACK SLOT.
1540                            .NLIST
1541                            .ENDC
1542                            .ENDM
1543                            .LIST
1544                                    JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
1545                            .NLIST
1546                            .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.
                        ;*************************************************************************

003772                  PREG05:                 ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
003772  010446                  MOV     R4,-(SP)        ;SAVE R4
003774  010346                  MOV     R3,-(SP)        ;SAVE R3
003776  010246                  MOV     R2,-(SP)        ;SAVE R2
004000  010146                  MOV     R1,-(SP)        ;SAVE R1
004002  010046                  MOV     R0,-(SP)        ;SAVE R0
004004  010546                  MOV     R5,-(SP)        ;PUSH RETURN PC ON TOP OF STACK
004006  016605  000014          MOV     R5SLOT(SP),R5   ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS

004012  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.
                                ;---

004014  012605          PREGRT:: MOV    (SP)+,R5        ;PUT RETURN PC IN R5.
004016  012600                  MOV     (SP)+,R0        ;RESTORE R0.
004020  012601                  MOV     (SP)+,R1        ;RESTORE R1.
004022  012602                  MOV     (SP)+,R2        ;RESTORE R2.
004024  012603                  MOV     (SP)+,R3        ;RESTORE R3.
004026  012604                  MOV     (SP)+,R4        ;RESTORE R4.

004030  000205                  RTS     R5              ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05.
                                                          ;RESTORING R5 IN THE PROCESS.
```

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


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

                        ;
                        ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                        ;
                              DEVTYP  <DHU-11>
004032                                                          L$DVTYP::
004032                                                                .ASCIZ  /DHU-11/
004032  104     110     125
004035  055     061     061
004040  000

                                                                .EVEN


                        ; TEST DESCRIPTION
                        ;
                              DESCRIPT        <DHU-11 FUNC TST PART1>
004042                                                          L$DESC::
004042                                                                .ASCIZ  /DHU-11 FUNC TST PAR
004042  104     110     125
T1/
004045  055     061     061
004050  040     106     125
004053  116     103     040
004056  124     123     124
004061  040     120     101
004064  122     124     061
004067  000

                                                                .EVEN

                              .EVEN
```

```
    1642
    1643                        ;*****************************************************************************************
*
    1644                        ;
    1645                        ;               FVTA.FMT
    1646                        ;
    1647                        ;*****************************************************************************************
    1648
    1649
    1650
    1651                        ;
    1652                        ; FORMAT STATEMENTS USED IN PRINT CALLS
    1653                        ;
    1654
    1665
    1666
```

```
                               ;**********************************************************************
                               ;
                               ;              FVTA.MSG
                               ;
                               ;**********************************************************************
**


              .NLIST BIN
              .SBTTL  GLOBAL MESSAGE AREA
                        ; ********** FORMAT STATEMENTS **********
       004070 MFUNIT:: .ASCIZ /%N%A TESTING UNIT :%D4%N/
       004121 EF0503:: .ASCIZ /%T%N/
       004126 EF0505:: .ASCIZ /%A    %D5%A ILLEGAL INTERRUPTS RECEIVED.%N/
       004201 EF1401:: .ASCIZ /%N%A ROM VERSION NUMBERS: PROC_1 = %D2%A(D)   PROC_2 = %D2%A(D)%N/
       004303 EF1402:: .ASCIZ /%T%A ROM VERSION NUMBER %T%N/
       004340 EF1601:: .ASCIZ /%A  %T%A, TEST ABORTED %N/
       004372 EF1602:: .ASCIZ /%A     EXPECTED DATA: %O6%A (O).%N/
       004434 EF1603:: .ASCIZ /%A     ACTUAL DATA:   %O6%A (O).%N/
       004476 EF1604:: .ASCIZ /%A  BAD BIT(S) IN DEVICE %T%A REGISTER FOR LINE %D2%A (D).%N/
       004573 EF3001:: .ASCIZ /%A     EXPECTED OR CORRECT VALUE: %O3%N/
       004642 EF3002:: .ASCIZ /%A     ACTUAL OR MEASURED  VALUE: %O3%N/
       004711 EF9006:: .ASCIZ /%A  %T%A %D2%A(D)%N/
       004735 EF9010:: .ASCIZ /%A     NUMBER OF ERRORS DETECTED ON LINE %D2%A(D) IS %D5%A(D)%N/
       005034 EF9016:: .ASCIZ /%A  UNEXPECTED %T%A FOR LINE %D2%A(D) IN FIFO AFTER RESET:%N/
       005131 EF9017:: .ASCIZ /%A    %T%A (WITH ERROR FLAGS) IS %O6%A(O)%N/
       005205 EF9018:: .ASCIZ /%A  %T%A IN SELFTEST CODE FIFO SLOT FOR LINE %D2/
       005265          .ASCIZ /%A(D) AFTER RESET.%N/
       005312 EF9301:: .ASCIZ /%A %T%D2%A(D),  BMP CODE REPORTED :%O3%A(O)%N/
       005370 EF9302:: .ASCIZ /%A  OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)%N/

                       ;********** ERROR MESSAGES ************
       005470 DR00MG:: .ASCIZ /CSR/
       005474 DR02MG:: .ASCIZ /RBUF/
       005501 DR04MG:: .ASCIZ /LPR/
       005505 DR06MG:: .ASCIZ /FIFOSIZE,STAT/
       005523 DR10MG:: .ASCIZ /LNCTRL/
       005532 DR12MG:: .ASCIZ /TBUFFAD1/
       005543 DR14MG:: .ASCIZ /TBUFFAD2/
       005554 DR16MG:: .ASCIZ /TBUFFCT/
       005564 EM0103:: .ASCIZ / DEVICE REGISTER ADDRESS TEST FAILED./
       005632 EM0201:: .ASCIZ / MASTER RESET TEST FAILED./
       005665 EM0202:: .ASCIZ /  MASTER RESET BIT DID NOT CLEAR AFTER BOARD RESET./
       005751          .ASCIZ /     WAITED 5 SECONDS.  BIT DEFECTIVE OR FIRMWARE HUNG./
       006040 EM0203:: .ASCIZ /  MASTER RESET BIT CLEAR IMMEDIATELY AFTER BOARD RESET./
       006130          .ASCIZ /     BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
       006203 EM0204:: .ASCIZ \  MR BIT WENT CLEAR WITHIN 1/2 SECOND OF BOARD RESET.\
       006271          .ASCIZ /     BIT DEFECTIVE OR SELFTEST WAS (INCORRECTLY) SKIPPED./
       006362 EM0301:: .ASCIZ /MASTER RESET (SKIP SELFTEST) TEST FAILED./
       006434 EM0302:: .ASCIZ /  MR BIT CLR WITHIN 10 MILISECOND AFTER BOARD RESET./
       006521          .ASCIZ /     BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
       006574 EM0303:: .ASCIZ \  MR BIT WENT CLEAR 1/5 TO 5 SECONDS AFTER RESET.\
       006656          .ASCIZ /     SELFTEST DID NOT GET SKIPPED (SHOULD HAVE BEEN SKIPPED)./
       006753 EM0401:: .ASCIZ /RBUF REGISTER RX CHARACTER FIELD TEST FAILED./
       007031 EM0402:: .ASCIZ /  IMPROPER CODE FOUND IN RX FIFO AFTER DUT RESET./
       007113          .ASCIZ /     EXPECTED: SELFTEST CODE,   ACTUAL: IMPROPER CODE./
       007201 EM0501:: .ASCIZ /RBUF REGISTER ERROR FLAGS FIELD TEST FAILED/
       007255 EM0502:: .ASCIZ /  RX ERROR FLAG(S) FOUND CLEAR ON SELFTEST CODE./
```

```
1732 007336            .ASCIZ /    EXPECTED: ALL ERROR FLAGS SET,  ACTUAL: FLAG(S) CLEAR./
1733 007431  EM0525:: .ASCIZ /  RX INTERRUPT(S) RECEIVED WITH RX INTERRUPTS DISABLED./
1734 007521  EM0526:: .ASCIZ /  TX INTERRUPT(S) RECEIVED WITH TX INTERRUPTS DISABLED./
1735 007611  EM0601:: .ASCIZ /CSR RX.DATA.AVAIL BIT TEST FAILED/
1736 007653  EM0602:: .ASCIZ /  RX.DATA.AVAIL BIT FOUND CLEAR AFTER RESET COMPLETION./
1737 007743            .ASCIZ /    EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
1738 010033  EM0603:: .ASCIZ /  RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO./
1739 010125            .ASCIZ /    600 CHARS READ FROM FIFO WITHOUT R.D.A BIT CLEARING./
1740 010216  EM0701:: .ASCIZ /RBUF RX.DATA.VALID BIT TEST FAILED/
1741 010261  EM0702:: .ASCIZ /  RX.DATA.VALID BIT FOUND CLEAR AFTER RESET COMPLETION./
1742 010351            .ASCIZ /    EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
1743 010441  EM0703:: .ASCIZ /  RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO./
1744 010533            .ASCIZ /    600 CHARS READ FROM FIFO WITHOUT R.D.V BIT CLEARING./
1745 010624  EM0801:: .ASCIZ /RBUF RX.LINE.NUMBER FIELD TEST FAILED/
1746 010672  EM0802:: .ASCIZ /  LINE NUMBER WRONG ON A SELFTEST CODE./
1747 010742  EM0901:: .ASCIZ /CHECK FOR BMP_CODES TEST FAILED/
1748 011002  EM0902:: .ASCIZ /UNEXPECTED BMP CODES FOUND./
1749 011036  EM1001:: .ASCIZ /SKIP SELF-TEST TEST FAILED/
1750 011071  EM1002:: .ASCIZ /  SKIP SELF-TEST TOOK TOO LONG TO COMPLETE, > 50 MS./
1751 011156  EM1003:: .ASCIZ /  SKIP SELF-TEST COMPLETED TOO SOON, < 10 MS./
1752 011234  EM1101:: .ASCIZ /DIAGNOSTIC FAIL (SKP SELFTEST) TEST FAILED/
1753 011307  EM1201:: .ASCIZ /SELF-TEST TEST FAILED/
1754 011335  EM1202:: .ASCIZ /  SELF-TEST TOOK TOO LONG TO COMPLETE, > 5 SECONDS./
1755 011421  EM1203:: .ASCIZ \  SELF-TEST COMPLETED TOO SOON, < 1/2 SECOND.\
1756 011477  EM1204:: .ASCIZ /  SELF-TEST DID NOT EXECUTE/
1757 011533  EM1205:: .ASCIZ /  DIAG_FAIL BIT BAD/
1758 011557  EM1301:: .ASCIZ /FAIL SELF-TEST TEST FAILED/
1759 011612  EM1302:: .ASCIZ / SELF-TEST ERROR REPORTING BAD/
1760 011651  EM1401:: .ASCIZ /ROM VERSION_NUMBER TEST FAILED/
1761 011710  EM1402:: .ASCIZ /  FIFO EMPTY, ONE OR MORE ROM VERSION_NUMBERS MISSING/
1762 011776  EM1403:: .ASCIZ /  ROM VERSION_NUMBER FOUND OUT OF SEQUENCE/
1763 012051  EM1404:: .ASCIZ /  ONE OR MORE ROM VERSION_NUMBERS MISSING/
1764 012123  EM1405:: .ASCIZ /    PROC_1/
1765 012136  EM1406:: .ASCIZ /    PROC_2/
1766 012151  EM1407:: .ASCIZ /NOT FOUND/
1767 012163  EM1408:: .ASCIZ /FOUND/
1768 012171  EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
1769 012254  EM1604:: .ASCIZ \DEVICE REGISTER WORD READ/WRITE TEST FAILED\
1770 012330  EM1701:: .ASCIZ \DEVICE REGISTER WORD READ/MODIFY/WRITE TEST FAILED\
1771 012413  EM1801:: .ASCIZ \DEVICE REGISTER BYTE READ/WRITE TEST FAILED\
1772 012467  EM1901:: .ASCIZ \DEVICE REGISTER BYTE READ/MODIFY/WRITE TEST FAILED\
1773 012552  EM2001:: .ASCIZ /DEVICE STAT REGISTER ID BIT TEST FAILED/
1774 012622  EM2002:: .ASCIZ /ID BIT BAD.  EXPECTED: SET, ACTUAL: CLEAR./
1775 012675  EM2301:: .ASCIZ /TX_ENABLE (INACTIVE) BIT TEST FAILED/
1776 012742  EM2302:: .ASCIZ /  TX_ENABLE BIT BAD ON LINE: /
1777 013000  EM2401:: .ASCIZ /TX_ENABLE (ACTIVE) BIT TEST FAILED/
1778 013043  EM2601:: .ASCIZ /RECEIVE INTERRUPT TEST FAILED/
1779 013101  EM2602:: .ASCIZ /  NO RX INT GENERATED (DATA_VALID SET, RX INTS ENABLED)./
1780 013172  EM2603:: .ASCIZ /  NO RX INT GENERATED (NO CODES IN FIFO AFTER RESET)./
1781 013260  EM2604:: .ASCIZ /  NO RX INT GENERATED (RX_DATA_AVAIL CLR, RX INTS ENABLED)./
1782 013354  EM2605:: .ASCIZ /  RX INTERRUPT GENERATED WITH RX_DATA_AVAIL CLEAR./
1783 013437  EM2606:: .ASCIZ /TRANSMIT INTERRUPT TEST ERROR:/
1784 013476  EM2607:: .ASCIZ /  TX_ACTION SET REPEATEDLY AFTER BOARD RESET, NO DATA SENT./
1785 013572  EM2608:: .ASCIZ /  TX_ACTION STUCK SET AFTER BOARD RESET./
1786 013643  EM2609:: .ASCIZ /  TX INTERRUPT GENERATED WITH TX_ACTION CLEAR./
1787 013722  EM2610:: .ASCIZ /  NO TX INTERRUPT WITH TX_ACTION SET AND TX INTS ENABLED./
1788 014014  EM2611:: .ASCIZ /  TX_ACTION NOT SET AFTER CHARS SENT ON ALL LINES./
```

```
1789 014077  EM2612:: .ASCIZ /  NO RX INT GENERATED (RX_DATA_AVAIL SET, RX INTS ENABLED)./
1790 014173  EM3001:: .ASCIZ /INTERRUPT BR LEVEL TEST FAILED/
1791 014232  EM3002:: .ASCIZ /  NO RX_DATA_AVAIL FROM SELFTEST CODES IN FIFO AFTER RESET./
1792 014326  EM3003:: .ASCIZ /  TX INTERRUPT GENERATED AT WRONG BR LEVEL:/
1793 014402  EM3004:: .ASCIZ /  RX INTERRUPT GENERATED AT WRONG BR LEVEL:/
1794 014456  EM3005:: .ASCIZ /  TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS RX INT./
1795 014550  EM3101:: .ASCIZ /DIAGNOSTIC FIELD (BMP) TEST FAILED/
1796 014613  EM3102:: .ASCIZ /  DIAGNOSTIC FIELD (BMP REQUEST) BAD ON LINE: /
1797 014672  EM9010:: .ASCIZ /ACTUAL OR MEASURED /
1798 014716  EM9014:: .ASCIZ /SUMMARY REPORTS FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:/
1799 015012  EM9017:: .ASCII /  FIFO WILL NOT PURGE (DATA.VALID STUCK SET),/
1800 015067          .ASCIZ / REMAINDER OF TEST SKIPPED./
1801 015123  EM9018:: .ASCIZ /NO CODE/
1802 015133  EM9019:: .ASCIZ /NON-SELFTEST/
1803 015150  EM9020:: .ASCIZ /SELFTEST ERROR CODE/
1804 015174  EM9022:: .ASCIZ /DATA CHARACTER/
1805 015213  EM9023:: .ASCIZ /MODEM STATUS CODE/
1806 015235  EM9024:: .ASCIZ /SELFTEST CODE/
1807 015253  EM9301:: .ASCIZ /BMP CODE REPORT/
1808 015273  EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
1809 015323  EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
1810 015370  EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
1811                 .EVEN
1812                          .LIST BIN
```

```
1814
1815
1816                    ;*****************************************************************************
1817                    ;
1818                    ;               FVTSKL2.P11
1819                    ;
1820                    ;*****************************************************************************
1821
1822
1823
1824                    .SBTTL  GLOBAL ERROR REPORT SECTION
1825
1826                    ;++
1827                    ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
1828                    ; USED BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
1829                    ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
1830                    ;--
```

```
1832                                    .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0101 -
1833                                    ;*************************************************************************
1834                                    ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1835                                    ;*      INFORMATION IF AN ERROR IS DETECTED IN TEST 1 (REGISTER ADDRESS
1836                                    ;*      ACCESS TEST). IF THE "EXTENDED ERROR INFO" OPTION HAS BEEN SELECTED
1837                                    ;*      THEN THIS SUBROUTINE WILL REPORT THE TYPE OF ACCESS (READ OR WRITE OR
1838                                    ;*      BOTH) WHICH CAUSED A BUS TIME-OUT TRAP (004 TRAP).A MESSAGE INDICATING
1839                                    ;*      THAT THE DHU MAY BE AT THE WRONG UNIBUS ADDRESS IS ALSO PRINTED.
1840                                    ;*
1841                                    ;* INPUTS:      R5 - ERROR FLAG WORD.
1842                                    ;*                   IF BIT 0 IS SET, A READ ERROR OCCURED.
1843                                    ;*                   IF BIT 1 IS SET, A WRITE ERROR OCCURED.
1844                                    ;*
1845                                    ;* OUTPUTS:     MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE.
1846                                    ;*
1847                                    ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER0101" AS THE MESSAGE POINTER
1848                                    ;*                      PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
1849                                    ;*
1850                                    ;* COMMENTS:
1851                                    ;*
1852                                    ;* SUBORDINATE ROUTINES USED: NONE.
1853                                    ;*************************************************************************
1854
1855 015444                                 BGNMSG  ER0101
       015444                                                                   ER0101::
1856 015444                                 SAVE                    ;SAVE THE GPR CONTENTS.
       015444  004567  166322                       JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
1857
1858 015450  012700  000100                  MOV     #BIT06,R0       ;SET-UP THE BIT MAP FOR 'REPORT EXT'D ERROR INFO'
1859 015454  046700  164540                  BIC     OPTION,R0       ;TRY AND CLEAR THE FLAG.
1860 015460  001036                         BNE     6$              ;EXIT IF OPTION NOT SELECTED.
1861                                    ;+
1862                                    ; REPORT EXTENDED ERROR INFOMATION
1863                                    ;-
1864
1865 015462  032705  000001                  BIT     #BIT0,R5        ;TEST FOR READ ERROR.
1866 015466  001410                         BEQ     2$              ;SKIP READ ERROR MSG IF NO READ ERROR.
1867 015470                                 PRINTB  #MSG1           ;PRINT READ ERROR MESSAGE.
       015470  012746  015562                                                   MOV     #MSG1,-(SP)
       015474  012746  000001                                                   MOV     #1,-(SP)
       015500  010600                                                           MOV     SP,R0
       015502  104414                                                           TRAP    C$PNTB
       015504  062706  000004                                                   ADD     #4,SP
1868 015510  032705  000002          2$:     BIT     #BIT1,R5        ;TEST FOR WRITE ERROR.
1869 015514  001410                         BEQ     4$              ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
1870 015516                                 PRINTB  #MSG2           ;PRINT WRITE ERROR MESSAGE.
       015516  012746  015640                                                   MOV     #MSG2,-(SP)
       015522  012746  000001                                                   MOV     #1,-(SP)
       015526  010600                                                           MOV     SP,R0
       015530  104414                                                           TRAP    C$PNTB
       015532  062706  000004                                                   ADD     #4,SP
1871 015536                                 4$:     PRINTX  #MSG3           ;SUGGEST THAT DHU MAY BE AT WRONG ADDRESS.
       015536  012746  015717                                                   MOV     #MSG3,-(SP)
       015542  012746  000001                                                   MOV     #1,-(SP)
       015546  010600                                                           MOV     SP,R0
       015550  104415                                                           TRAP    C$PNTX
       015552  062706  000004                                                   ADD     #4,SP
```

```
1872 015556                          6$:     PASS                    ;RESTORE THE GPR CONTENTS.
     015556  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
1873 015560                          ENDMSG
     015560                                                          L10002:
     015560  104423                                                          TRAP    C$MSG
1874
1875 015562      045     101     102  MSG1::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.%N/
     015565      125     123     040
     015570      124     111     115
     015573      105     055     117
     015576      125     124     040
     015601      124     122     101
     015604      120     040     103
     015607      101     125     123
     015612      105     104     040
     015615      102     131     040
     015620      122     105     101
     015623      104     040     101
     015626      124     124     105
     015631      115     120     124
     015634      056     045     116
     015637      000
1876 015640      045     101     102  MSG2::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.%N/
     015643      125     123     040
     015646      124     111     115
     015651      105     055     117
     015654      125     124     040
     015657      124     122     101
     015662      120     040     103
     015665      101     125     123
     015670      105     104     040
     015673      102     131     040
     015676      127     122     111
     015701      124     105     040
     015704      101     124     124
     015707      105     115     120
     015712      124     056     045
     015715      116     000
1877 015717      045     101     104  MSG3::  .ASCIZ  /%ADHU MAY BE AT THE WRONG UNIBUS ADDRESS.%N%N/
     015722      110     125     040
     015725      115     101     131
     015730      040     102     105
     015733      040     101     124
     015736      040     124     110
     015741      105     040     127
     015744      122     117     116
     015747      107     040     125
     015752      116     111     102
     015755      125     123     040
     015760      101     104     104
     015763      122     105     123
     015766      123     056     045
     015771      116     045     116
     015774      000
1878
1879                                        .EVEN
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0201 -
                        ;*****************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS 2 CONTIGUOUS
                        ;*      ASCII ERROR MESSAGES.  THE ADDRESS OF THE FIRST MESSAGE IS PASSED
                        ;*      AS AN INPUT PARAMETER AND THE ADDRESS OF THE SECOND IS FOUND BY
                        ;*      SEARCHING FOR THE END OF THE FIRST MESSAGE. THE MESSAGES ARE ONLY
                        ;*      PRINTED IF EXT'D ERROR REPORTING HAS BEEN REQUESTED.
                        ;*
                        ;* INPUTS:      R1 - ADDRESS OF THE FIRST MESSAGE TO PRINT.
                        ;*
                        ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    LOAD THE ADDRESS OF THE FIRST MESSAGE IN R1.
                        ;*                      INCLUDE THE LABEL "ER0201" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
                        ;*              THE SECOND MESSAGE SHOULD FOLLOW THE FIRST ONE IN THE PROGRAM
                        ;*              MEMORY.  EACH MESSAGE SHOULD BE DEFINED USING .ASCIZ
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*****************************************************************************

                              BGNMSG  ER0201
016776                                                                  ER0201::
                              SAVE                      ;SAVE THE GPR CONTENTS.
016776  004567  165770              JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

016002  012700  000100        MOV     #BIT06,R0       ;TRY TO CLEAR THE
016006  046700  164206        BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
016012  001025                BNE     4$              ;EXIT IF FLAG NOT SET.

016014  010102                MOV     R1,R2
016016  105722          2$:   TSTB    (R2)+           ;CHECK FOR A ZERO BYTE (END OF MESSAGE).
016020  001376                BNE     2$              ;LOOP UNTIL NEXT MESSAGE IS FOUND.

016022                        PRINTB  #EF0503,R1      ;PRINT THE FIRST MESSAGE.
016022  010146                                                  MOV     R1,-(SP)
016024  012746  004121                                         MOV     #EF0503,-(SP)
016030  012746  000002                                         MOV     #2,-(SP)
016034  010600                                                 MOV     SP,R0
016036  104414                                                 TRAP    C$PNTB
016040  062706  000006                                         ADD     #6,SP
016044                        PRINTB  #EF0503,R2      ;PRINT THE SECOND MESSAGE.
016044  010246                                                  MOV     R2,-(SP)
016046  012746  004121                                         MOV     #EF0503,-(SP)
016052  012746  000002                                         MOV     #2,-(SP)
016056  010600                                                 MOV     SP,R0
016060  104414                                                 TRAP    C$PNTB
016062  062706  000006                                         ADD     #6,SP

016066          4$:   PASS                      ;RESTORE THE GPR CONTENTS.
016066  004736                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.

016070                        ENDMSG
016070                                                          L10003:
016070  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, PROVIDED
                        ;*      EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
                        ;*
                        ;* 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.
                        ;*************************************************************************

016072                          BGNMSG  ER0503
016072                                                                  ER0503::
016072  012700  000100                 MOV     #BIT06,R0       ;TRY TO CLEAR THE
016076  046700  164116                 BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
016102  001011                       BNE     2$              ;EXIT IF FLAG NOT SET.

016104                          PRINTB  #EF0503,R1      ;PRINT THE MESSAGE.
016104  010146                                                          MOV     R1,-(SP)
016106  012746  004121                                                  MOV     #EF0503,-(SP)
016112  012746  000002                                                  MOV     #2,-(SP)
016116  010600                                                          MOV     SP,R0
016120  104414                                                          TRAP    C$PNTB
016122  062706  000006                                                  ADD     #6,SP

016126                  2$:     ENDMSG
016126                                                          L10004:
016126  104423                                                          TRAP    C$MSG
```

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

016130                          BGNMSG  ER0504
016130                                                                  ER0504::

016130  012700  000100           MOV     #BIT06,R0       ;TRY TO CLEAR THE
016134  046700  164060           BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
016140  001022                  BNE     2$              ;EXIT IF FLAG NOT SET.

016142                          PRINTB  #EF0503,R1      ;PRINT THE FIRST LINE OF THE MESSAGE.
016142  010146                                                  MOV     R1,-(SP)
016144  012746  004121                                          MOV     #EF0503,-(SP)
016150  012746  000002                                          MOV     #2.-(SP)
016154  010600                                                  MOV     SP,R0
016156  104414                                                  TRAP    C$PNTB
016160  062706  000006                                          ADD     #6,SP
016164                          PRINTX  #EF0505,R2      ;PRINT THE NUMBER OF INTS RECEIVED.
016164  010246                                                  MOV     R2,-(SP)
016166  012746  004126                                          MOV     #EF0505,-(SP)
016172  012746  000002                                          MOV     #2.-(SP)
016176  010600                                                  MOV     SP,R0
016200  104415                                                  TRAP    C$PNTX
016202  062706  000006                                          ADD     #6,SP

016206                  2$:     ENDMSG
016206                                                  L10005:
016206  104423                                                  TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER1401 -
                        ;*************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
                        ;*      INFORMATION (IF REQUESTED DURING THE SOFTWARE QUESTIONS) 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.
                        ;*************************************************************************

 016210                         BGNMSG  ER1401
 016210                                                                 ER1401::

                         MOV    #BIT06,R0       ;TRY TO CLEAR THE
                         BIC    OPTION,R0       ;EXT'D ERROR REPORTING FLAG
                        BNE     60$             ;EXIT IF FLAG NOT SET.

 016222                 PRINTB  #EF0503,R1      ;REPORT THE ERROR MESSAGE PASSED IN.
 016222 010146                                                          MOV     R1,-(SP)
 016224 012746  004121                                                  MOV     #EF0503,-(SP)
 016230 012746  000002                                                  MOV     #2,-(SP)
 016234 010600                                                          MOV     SP,R0
 016236 104414                                                          TRAP    C$PNTB
 016240 062706  000006                                                  ADD     #6,SP

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

                         MOV    #99.,R5         ;GET INVALID ROM NUMBER.
                         MOV    #EM1405,R1      ;SELECT PROC_1 MESSAGE.
                         MOV    #EM1407,R2      ;SELECT THE "NOT FOUND" MESSAGE.
                         CMPB   R3,R5           ;CHECK PROC_1 ROM VERSION NUMBER.
                        BEQ     2$              ;GO REPORT PROC_1 CODE NOT FOUND.
                         MOV    #EM1408,R2      ;SELECT "FOUND" MESSAGE.
                2$:     JSR     PC,50$          ;GO REPORT MESSAGE.

                         MOV    #EM1406,R1      ;SELECT PROC_2 MESSAGE.
                         MOV    #EM1407,R2      ;SELECT THE "NOT FOUND" MESSAGE.
                         CMPB   R4,R5           ;CHECK PROC_2 ROM VERSION NUMBER.
                        BEQ     4$              ;GO REPORT PROC_2 CODE NOT FOUND.
                         MOV    #EM1408,R2      ;SELECT "FOUND" MESSAGE.
                4$:     JSR     PC,50$          ;GO REPORT THE MESSAGE.
                        BR      60$             ;EXIT.
```

```
2033
2034 016322                        50$:    PRINTX  #EF1402,R1,R2   ;REPORT THE MESSAGE.
     016322  010246                                                          MOV     R2,-(SP)
     016324  010146                                                          MOV     R1,-(SP)
     016326  012746  004303                                                  MOV     #EF1402,-(SP)
     016332  012746  000003                                                  MOV     #3,-(SP)
     016336  010600                                                          MOV     SP,R0
     016340  104415                                                          TRAP    C$PNTX
     016342  062706  000010                                                  ADD     #10,SP
2035 016346  000207                         RTS     PC              ;RETURN.
2036 016350                        60$:    ENDMSG
     016350                                                     L10006:
     016350  104423                                                         TRAP    C$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, PROVIDED EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
                        ;*      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
                        ;*************************************************************************
016352                          BGNMSG  ER1601
016352                                                                  ER1601::
016352  012700  000100                  MOV     #BIT06,R0       ;TRY TO CLEAR THE
016356  046700  163636                  BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
016362  001036                        BNE     2$              ;EXIT IF FLAG NOT SET.
016364  016304  002256                  MOV     RMATBB(R3),R4   ;FETCH ADDRESS OF REGISTER NAME MESSAGE.
016370                                  PRINTB  #EF1604,R4,R5   ;REPORT BASIC MESSAGE (REG NAME AND LINE #).
016370  010546                                                          MOV     R5,-(SP)
016372  010446                                                          MOV     R4,-(SP)
016374  012746  004476                                                  MOV     #EF1604,-(SP)
016400  012746  000003                                                  MOV     #3,-(SP)
016404  010600                                                          MOV     SP,R0
016406  104414                                                          TRAP    C$PNTB
016410  062706  000010                                                  ADD     #10,SP
016414                                  PRINTX  #EF1602,R2      ;PRINT THE EXPECTED DATA.
016414  010246                                                          MOV     R2,-(SP)
016416  012746  004372                                                  MOV     #EF1602,-(SP)
016422  012746  000002                                                  MOV     #2,-(SP)
016426  010600                                                          MOV     SP,R0
016430  104415                                                          TRAP    C$PNTX
016432  062706  000006                                                  ADD     #6,SP
016436                                  PRINTX  #EF1603,R1      ;PRINT THE ACTUAL DATA.
016436  010146                                                          MOV     R1,-(SP)
016440  012746  004434                                                  MOV     #EF1603,-(SP)
016444  012746  000002                                                  MOV     #2,-(SP)
016450  010600                                                          MOV     SP,R0
016452  104415                                                          TRAP    C$PNTX
016454  062706  000006                                                  ADD     #6,SP
016460                          2$:     ENDMSG
016460                                                          L10007:
```

```
016460  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, PROVIDED EXTENDED ERROR INFOMATION HAS BEEN
                        ;*      REQUESTED, OTHERWISE ONLY A "TEST FAILURE" MESSAGE WILL BE PRINTED.
                        ;*
                        ;* 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.
                        ;*************************************************************************
                                BGNMSG  ER1603
                                                                                ER1603::
                                SAVE                            ;SAVE THE CONTENTS OF THE GPRS.
016462  004567  165304                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

016466  012700  000100          MOV     #BIT06,R0               ;TRY TO CLEAR THE
016472  046700  163522          BIC     OPTION,R0               ;EXT'D ERROR REPORTING FLAG
016476  001024                  BNE     2$                      ;EXIT IF FLAG NOT SET.


016500                          PRINTB #EF0503,R1               ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
016500  010146                                                          MOV     R1,-(SP)
016502  012746  004121                                                  MOV     #EF0503,-(SP)
016506  012746  000002                                                  MOV     #2,-(SP)
016512  010600                                                          MOV     SP,R0
016514  104414                                                          TRAP    C$PNTB
016516  062706  000006                                                  ADD     #6,SP

016522  016702  165240          MOV     ERRMSG,R2               ;GET THE "TEST MESSAGE".
016526                          PRINTB #EF1601,R2               ;PRINT "TEST ABORTED" MESSAGE.
016526  010246                                                          MOV     R2,-(SP)
016530  012746  004340                                                  MOV     #EF1601,-(SP)
016534  012746  000002                                                  MOV     #2,-(SP)
016540  010600                                                          MOV     SP,R0
016542  104414                                                          TRAP    C$PNTB
016544  062706  000006                                                  ADD     #6,SP

016550                  2$:     PASS                    ;RESTORE THE CONTENTS OF THE GPRS.
016550  004736                  JSR     PC,@(SP)+                       ;RETURN TO PREG05 SUBRT.
016552                          ENDMSG
016552                                                          L10010:
016552  104423                                                          TRAP    C$MSG
```

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

016554                          BGNMSG  ER3001
016554                                                                  ER3001::
016554  012700  000100                  MOV     #BIT06,R0       ;TRY TO CLEAR THE
016560  046700  163434                  BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
016564  001033                  BNE     2$                      ;EXIT IF FLAG NOT SET.

016566                          PRINTB  #EF0503,R1      ;PRINT THE FIRST LINE OF THE MESSAGE.
016566  010146                                                          MOV     R1,-(SP)
016570  012746  004121                                                  MOV     #EF0503,-(SP)
016574  012746  000002                                                  MOV     #2,-(SP)
016600  010600                                                          MOV     SP,R0
016602  104414                                                          TRAP    C$PNTB
016604  062706  000006                                                  ADD     #6,SP
016610                          PRINTX  #EF3001,R5      ;REPORT EXPECTED BR LEVEL.
016610  010546                                                          MOV     R5,-(SP)
016612  012746  004573                                                  MOV     #EF3001,-(SP)
016616  012746  000002                                                  MOV     #2,-(SP)
016622  010600                                                          MOV     SP,R0
016624  104415                                                          TRAP    C$PNTX
016626  062706  000006                                                  ADD     #6,SP
016632                          PRINTX  #EF3002,R4      ;REPORT ACTUAL BR LEVEL.
016632  010446                                                          MOV     R4,-(SP)
016634  012746  004642                                                  MOV     #EF3002,-(SP)
016640  012746  000002                                                  MOV     #2,-(SP)
016644  010600                                                          MOV     SP,R0
016646  104415                                                          TRAP    C$PNTX
016650  062706  000006                                                  ADD     #6,SP

016654                  2$:     ENDMSG
016654                                                          L10011:
016654  104423                                                          TRAP    C$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, PROVIDED EXTENDED ERROR REPORTING HAS
                        ;*      BEEN REQUESTED BY THE OPERATOR.
                        ;*
                        ;* 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.
                        ;*****************************************************************************

016656                          BGNMSG  ER9004
016656                                                                  ER9004::

016656  012700  000100                   MOV     #BIT06,R0       ;TRY TO CLEAR THE
016662  046700  163332                   BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
016666  001040                          BNE     6$              ;EXIT IF FLAG NOT SET.

016670                                  PRINTB  #EF0503,#EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
016670  012746  014716                                                  MOV     #EM9014,-(SP)
016674  012746  004121                                                  MOV     #EF0503,-(SP)
016700  012746  000002                                                  MOV     #2,-(SP)
016704  010600                                                          MOV     SP,R0
016706  104414                                                          TRAP    C$PNTB
016710  062706  000006                                                  ADD     #6,SP
016714  005002                          CLR     R2              ;CLEAR THE LINE COUNTER.
016716  016703  163534                   MOV     ERSMRF,R3       ;GET THE ERROR SUMMARY FLAGS.
016722  005004                          CLR     R4              ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
016724  000241          2$:             CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
016726  006003                          ROR     R3              ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
016730  103013                          BCC     4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
016732                                  PRINTX  #EF9010,R2,ERCNTB(R4)
016732  016446  002460                                                  MOV     ERCNTB(R4),-(SP)
016736  010246                                                          MOV     R2,-(SP)
016740  012746  004735                                                  MOV     #EF9010,-(SP)
016744  012746  000003                                                  MOV     #3,-(SP)
016750  010600                                                          MOV     SP,R0
016752  104415                                                          TRAP    C$PNTX
016754  062706  000010                                                  ADD     #10,SP
016760  012405          4$:             MOV     (R4)+,R5        ;INCREMENT THE LINE OFFSET BY 2.
016762  005202                          INC     R2              ;INCREMT THE LINE COUNTER.
016764  005703                          TST     R3              ;CHECK THE ERROR SUMMARY FLAGS.
016766  001356                          BNE     2$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.

016770          6$:             ENDMSG
016770                                                                  L10012:
```

```
016770  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. EXTENDED ERROR INFOMATION
                        ;*      IS GIVEN ONLY WHEN REQUESTED IN THE SOFTWARE QUESTIONS.
                        ;*
                        ;* 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.
                        ;*******************************************************************************

016772                          BGNMSG  ER9007
016772                                                                  ER9007::

016772  012700  000100                  MOV     #BIT06,R0       ;TRY TO CLEAR THE
016776  046700  163216                  BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
017002  001026                        BNE     2$              ;EXIT IF FLAG NOT SET.

017004  042703  177760                  BIC     #177760,R3      ;REMOVE ALL BUT LINE # BITS FROM LINE # WORD.
017010                                  PRINTB  #EF9018,R1,R3   ;REPORT SECONDARY ERROR MESSAGE.
017010  010346                                                          MOV     R3,-(SP)
017012  010146                                                          MOV     R1,-(SP)
017014  012746  005205                                                  MOV     #EF9018,-(SP)
017020  012746  000003                                                  MOV     #3,-(SP)
017024  010600                                                          MOV     SP,R0
017026  104414                                                          TRAP    C$PNTB
017030  062706  000010                                                  ADD     #10,SP
017034                                  PRINTX  #EF9017,R1,R2   ;REPORT THE ACTUAL INCORRECT CODE.
017034  010246                                                          MOV     R2,-(SP)
017036  010146                                                          MOV     R1,-(SP)
017040  012746  005131                                                  MOV     #EF9017,-(SP)
017044  012746  000003                                                  MOV     #3,-(SP)
017050  010600                                                          MOV     SP,R0
017052  104415                                                          TRAP    C$PNTX
017054  062706  000010                                                  ADD     #10,SP

017060                          2$:     ENDMSG
017060                                                          L10013:
017060  104423                                                          TRAP    C$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. THE ADDITIONAL ERROR IS REPORTED ONLY IF REQUESTED
                        ;*      DURING THE SOFTWARE QUESTIONS.
                        ;*
                        ;* 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.
                        ;*****************************************************************************

017062                          BGNMSG  ER9008
017062                                                                  ER9008::

017062  012700  000100                   MOV     #BIT06,R0       ;TRY TO CLEAR THE
017066  046700  163126                   BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
017072  001030                           BNE     2$              ;EXIT IF FLAG NOT SET.

                        ;*
                        ; EXTRACT THE LINE NUMBER FROM THE INCORRECT CODE OR CHARACTER WHICH WAS READ
                        ;    FROM THE SELFTEST CODE FIFO SLOT.
                        ;-
017074  010203                           MOV     R2,R3
017076  000303                           SWAB    R3
017100  042703  177760                   BIC     #177760,R3      ;CALCULATE LINE NUMBER OF CODE.
017104                                  PRINTB  #EF9016,R1,R3   ;REPORT TYPE OF INCORRECT CODE FOUND.
017104  010346                                                                  MOV     R3,-(SP)
017106  010146                                                                  MOV     R1,-(SP)
017110  012746  005034                                                          MOV     #EF9016,-(SP)
017114  012746  000003                                                          MOV     #3,-(SP)
017120  010600                                                                  MOV     SP,R0
017122  104414                                                                  TRAP    C$PNTB
017124  062706  000010                                                          ADD     #10,SP
017130                                  PRINTX  #EF9017,R1,R2   ;REPORT THE ACTUAL INCORRECT CODE.
017130  010246                                                                  MOV     R2,-(SP)
017132  010146                                                                  MOV     R1,-(SP)
017134  012746  005131                                                          MOV     #EF9017,-(SP)
017140  012746  000003                                                          MOV     #3,-(SP)
017144  010600                                                                  MOV     SP,R0
017146  104415                                                                  TRAP    C$PNTX
017150  062706  000010                                                          ADD     #10,SP

017154                  2$:     ENDMSG
017154                                                          L10014:
017154  104423                                                                  TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9101 -
                        ;*************************************************************************
                        ;*      THIS IS A GENERAL ERROR REPORTING SUBROUTINE WHICH REPORTS A MESSAGE
                        ;*      WHICH TAKES A SINGLE, 2 DIGIT DECIMAL ARGUMENT AFTER THE END OF AN
                        ;*      ASCII MESSAGE.
                        ;*
                        ;* INPUTS:      R1 - VALUE TO BE PRINTED AFTER MSG AS 2 DECIMAL DIGITS.
                        ;*              R2 - ADDRESS OF MESSAGE TO PRINT FIRST.
                        ;*
                        ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9101" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*************************************************************************

017156                          BGNMSG  ER9101
017156                                                                  ER9101::

017156  012700  000100                   MOV     #BIT06,R0       ;TRY TO CLEAR THE
017162  046700  163032                   BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
017166  001012                           BNE     2$              ;EXIT IF FLAG NOT SET.


017170                          PRINTB  #EF9006,R2,R1   ;REPORT THE STRING FOLLOWED BY THE NUMBER.
017170  010146                                                          MOV     R1,-(SP)
017172  010246                                                          MOV     R2,-(SP)
017174  012746  004711                                                  MOV     #EF9006,-(SP)
017200  012746  000003                                                  MOV     #3,-(SP)
017204  010600                                                          MOV     SP,R0
017206  104414                                                          TRAP    C$PNTB
017210  062706  000010                                                  ADD     #10,SP

017214                  2$:     ENDMSG
017214                                                          L10015:
017214  104423                                                          TRAP    C$MSG
```

```
2292                            .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9301 -
2293                            ;*****************************************************************************
2294                            ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ANY BMP CODES
2295                            ;*      THAT ARE FOUND IN THE BMP CODE QUEUE, TOGETHER WITH THE THE NUMBER OF
2296                            ;*      THE TEST THAT WAS EXECUTING AT THE TIME THE BMP CODE WAS LOGGED.
2297                            ;*      PROVIDED EXTENDED ERROR REPORTING HAS BEEN ENABLED.
2298                            ;*
2299                            ;* INPUTS:      R1 - THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
2300                            ;*              R2 - THE ADDRESS OF THE NEXT EMPTY CELL IN THE QUEUE.
2301                            ;*
2302                            ;* OUTPUTS:     THE TEST NUMBER FOLLOWED BY THE BMP CODE ARE PRINTED AT THE
2303                            ;*              OPERATOR CONSOLE.
2304                            ;*
2305                            ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9301" AS THE MESSAGE POINTER
2306                            ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2307                            ;*
2308                            ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2309                            ;*
2310                            ;* SUBORDINATE ROUTINES USED: NONE.
2311                            ;*****************************************************************************
2312
2313 017216                             BGNMSG  ER9301
   017216                                                                   ER9301::
2314 017216                             SAVE                    ;SAVE THE GPRS ON THE STACK.
   017216  004567  164550                       JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2315
2316 017222  012700  000100              MOV     #BIT06,R0       ;TRY TO CLEAR THE
2317 017226  046700  162766              BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
2318 017232  001064                      BNE     60$             ;EXIT IF FLAG NOT SET.
2319
2320 017234                             PRINTB  #EF0503,R1      ;REPORT UNEXPECTED BMP CODES FOUND.
   017234  010146                                                          MOV     R1,-(SP)
   017236  012746  004121                                                  MOV     #EF0503,-(SP)
   017242  012746  000002                                                  MOV     #2,-(SP)
   017246  010600                                                          MOV     SP,R0
   017250  104414                                                          TRAP    C$PNTB
   017252  062706  000006                                                  ADD     #6,SP
2321 017256  012703  002522              MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
2322 017262  012705  015273              MOV     #EM9302,R5      ;GET THE MESSAGE TO BE REPORTED.
2323 017266  012301              2$:     MOV     (R3)+,R1        ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
2324 017270  012304                      MOV     (R3)+,R4        ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
2325 017272  004767  000056              JSR     PC,50$          ;GO REPORT THE BMP CODE.
2326 017276  020302                      CMP     R3,R2           ;CHECK IF ALL CODES HAVE BEEN REPORTED.
2327 017300  103772                      BLO     2$              ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
2328                            ;+
2329                            ; CHECK IF OVERFLOW HAS OCCURRED.
2330                            ;  THE CONDITIONS FOR OVERFLOW ARE: THE POINTER CONTAINS THE ADDRESS OF THE
2331                            ;  LAST CELL IN THE QUEUE, AND A BMP CODE HAS ALREADY BEEN WRITTEN INTO THAT
2332                            ;  CELL.
2333                            ;-
2334 017302  020227  002716              CMP     R2,#BMPCQE-4    ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
2335 017306  001036                      BNE     60$             ;EXIT IF NOT AT THE LAST LOCATION.
2336 017310  005762  000002              TST     2(R2)           ;CHECK FOR A BMP CODE IN THE LAST CELL
2337 017314  001433                      BEQ     60$             ;EXIT IF NO OVERFLOW HAS OCCURED. CELL EMPTY.
2338 017316  012301                      MOV     (R3)+,R1        ;GET THE TEST NUMBER OFF THE QUEUE.
2339 017320  011304                      MOV     (R3),R4         ;GET THE BMP CODE OFF THE QUEUE.
2340 017322  012705  015323              MOV     #EM9303,R5      ;SELECT THE MESSAGE TO BE REPORTED.
```

```
2341 017326                                PRINTX  #EF9302          ;REPORT OVERFLOW CONDITION.
     017326  012746  005370                                                MOV     #EF9302,-(SP)
     017332  012746  000001                                                MOV     #1,-(SP)
     017336  010600                                                        MOV     SP,R0
     017340  104415                                                        TRAP    C$PNTX
     017342  062706  000004                                                ADD     #4,SP
2342 017346  004767  000002                JSR     PC,50$          ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
2343 017352  000414                        BR      60$             ;EXIT.
2344
2345 017354                          50$:  PRINTX  #EF9301,R5,R1,R4  ;PRINT THE MESSAGE.
     017354  010446                                                        MOV     R4,-(SP)
     017356  010146                                                        MOV     R1,-(SP)
     017360  010546                                                        MOV     R5,-(SP)
     017362  012746  005312                                                MOV     #EF9301,-(SP)
     017366  012746  000004                                                MOV     #4,-(SP)
     017372  010600                                                        MOV     SP,R0
     017374  104415                                                        TRAP    C$PNTX
     017376  062706  000012                                                ADD     #12,SP
2346 017402  000207                        RTS     PC              ;RETURN.
2347 017404                          60$:  PASS                    ;RESTORE THE GPR CONTENTS.
     017404  004736                                JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
2348
2349 017406                                ENDMSG
     017406                                                          L10016:
     017406  104423                                                        TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL SUBROUTINES SECTION
                        ;*****************************************************************************
                        ;
                        ;               FVTSKL3.P11
                        ;
                        ;*****************************************************************************


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

```
                                .SBTTL  GLOBAL SUBROUTINE                - ALTFLD -
                                ;++ ************************************************************************************
                                ;*              - ALTER DEVICE REGISTER FIELDS ROUTINE -
                                ;*      THIS SUBROUTINE ALTERS THE SPECIFIED FIELD OF THE SPECIFIED DEVICE
                                ;*      REGISTER FOR THE SPECIFIED LINES.  THIS ROUTINE CAN BE USED TO SET
                                ;*      OR CLEAR BITS WITHIN SELECTED FIELDS OF SELECTED REGISTERS.
                                ;*      USE EXAMPLES:   SET RX.BAUD.RATE FIELDS ON LINES 3 AND 6.
                                ;*                      CLEAR TX.DMA BITS ON ALL LINES.
                                ;*
                                ;* INPUTS:      R1 - ADDRESS OF THE REGISTERS TO ALTER.
                                ;*              R2 - BIT FIELDS SET TO DESIRED STATES.
                                ;*              R3 - BIT MAP OF LINES FOR WHICH TO ALTER REGISTER.
                                ;*              R4 - MASK OF BITS TO ALTER (1 INDICATES CHANGE BIT).
                                ;*              CSRA - CONTAINS THE ADDRESS OF THE DEVICE CSR.
                                ;*              IESTAT - SAVED STATES OF THE INTERRUPT ENABLE BITS.
                                ;*
                                ;* OUTPUTS:     DEVICE REGISTERS - SPECIFIED REGISTER FIELDS ALTERED.
                                ;*                  CSR IND.ADR.REG FIELD - DESTROYED.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,ALTFLD
                                ;*
                                ;* COMMENTS:    THIS ROUTINE READS THE SPECIFIED REGISTERS FOR ALL LINES
                                ;*              WITH NUMBERS LOWER THAN THE HIGHES SPECIFIED LINE.
                                ;*              THIS ROUTINE DOES NOT READ THE CSR.
                                ;*
                                ;* SUBROUTINES CALLED: NONE.
                                ;-- ************************************************************************************

017410                          ALTFLD:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017410  004567  164356                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

                                ;+
                                ; SET UP TO LOOP FOR EACH LINE:
                                ;    PREPARE THE WORD TO BE ORED INTO THE REGISTER CONTENTS.
                                ;    SET UP THE WORD TO WRITE INTO THE IND.ADR.REG FIELD OF THE CSR.
                                ;-
017414  010400                          MOV     R4,R0           ;CALCULATE THE NEW CONTENTS OF THE
017416  005100                          COM     R0              ;  REGISTER FIELDS WHICH ARE TO BE
017420  040002                          BIC     R0,R2           ;  ALTERRED BY THIS ROUTINE.
017422  016705  162656                  MOV     IESTAT,R5       ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.

                                ;+
                                ; LOOP ONCE FOR EACH LINE, ALTERING THE SPECIFIED FIELD IN THE SPECIFIED
                                ;    REGISTER IF THE LINE HAS BEEN SELECTED FOR ALTERING.
                                ; EXIT THE LOOP IF NO MORE LINES TO ALTER, OR IF WE HAVE ALTERED THE MAX
                                ;    ALLOWABLE NUMBER OF LINES (AS SPECIFIED BY NUMLNS).
                                ;-
017426  000241                          CLC                             ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
017430  006003                  2$:     ROR     R3                      ;GET THE LINE SELECT BIT FOR THIS LINE.
017432  103006                          BCC     4$                      ;SKIP SETUP IF LINE IS NOT SELECTED.
017434  010577  162576                  MOV     R5,@CSRA                ;SET DUT CSR IND.ADR.REG FIELD TO THIS LINE.
017440  011100                          MOV     (R1),R0                 ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
017442  040400                          BIC     R4,R0                   ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
017444  050200                          BIS     R2,R0                   ;OR IN THE NEW STATES OF THE FIELDS.
017446  010011                          MOV     R0,(R1)                 ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
017450  005205                  4$:     INC     R5                      ;SET LINE NUMBER TO THE NEXT LINE.
017452  005703                          TST     R3                      ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
017454  001365                          BNE     2$                      ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.
```

```
2422
2423 017456                        60$:  PASS                    ;RESTORE GPRS.
     017456  004736                            JSR   PC,@(SP)+   ;RETURN TO PREGOS SUBRT.
2424 017460  000207                        RTS   PC              ;RETURN TO CALLING ROUTNE.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - CALMSL -
                        ;++ ************************************************************************
                        ;*              - CALIBRATE MILLI SECOND LOOP COUNT SUBROUTINE -
                        ;*      THIS SUBROUTINE CALIBRATES THE TIMING LOOP WHICH IS USED IN THE MSLOOP
                        ;*      ROUTINE.  THIS SUBROUTINE CALCULATES A VALUE FOR THE MSLCNT VARIABLE
                        ;*      WHICH IS THE NUMBER OF SOFTWARE LOOPS WHICH TAKES 1 MS TO EXECUTE IN
                        ;*      THE MSLOOP ROUTINE.  THIS ROUTINE CALIBRATES THE COUNT BY USING THE
                        ;*      LINE TIME CLOCK (LTC), SO IF NO LTC IS AVAILABLE THE DEFAULT VALUE FOR
                        ;*      THE DELAY COUNT MUST BE USED.
                        ;*
                        ;*
                        ;* INPUTS:      MSLCNT - DEFAULT 1 MS DELAY LOOP COUNT VALUE, OR
                        ;*                       VALUE FROM PREVIOUS CALIBRATION.
                        ;*              MSTICK - NUMBER OF MS PER LTC CLOCK TICK.
                        ;*              TIMER1 - TIMER COUNTER CHANGED BY LTC INTERRUPT SERVICE RTN.
                        ;*              CLKHRZ - NUMBER OF LTC CLICKS PER SECOND (50 OR 60).
                        ;*
                        ;* OUTPUTS:     CARRY - SET IF LTC IS AVAILABLE, AND NEW CALIBRATION PERFORMED.
                        ;*              MSLCNT - NEW 1 MS DELAY LOOP COUNT VALUE IF LTC AVAILABLE, OR
                        ;*                       UNCHANGED IF NO LTC IS AVAILABLE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CALMSL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: UNSDIV,OOPS.
                        ;-- ************************************************************************

017462                  CALMSL:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
017462  004567  164304          JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
017466  005067  000210          CLR     62$             ;CLEAR THE 2ND TIME FLAG.
                        ;*
                        ; SYNCHRONIZE WITH THE LTC.
                        ;-
017472  012705  000001  2$:     MOV     #1,R5           ;SET OUTER LOOP COUNTER TO 1 LOOP.
                                                        ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE <**
                                                        ;FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <**
017476  005000          CLR     R0              ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
017500  012767  000001  162632  MOV     #1,TIMER1       ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
017506  005767  162626  4$:     TST     TIMER1          ;CHECK FOR COUNTER HAVING GONE TO ZERO.
017512  001410          BEQ     6$              ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
017514  005200          INC     R0              ;COUNT THIS ITERATION OF THE INNER LOOP.
017516  001373          BNE     4$              ;LOOP IF COUNTER HAS NOT TURNED OVER.
017520  005305          DEC     R5              ;DECREMENT THE INNER LOOP COUNTER.
017522  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.
                        ;-
017524  005067  162606          CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
017530  000241          CLC                             ;INDICATE FAILURE FOR RETURN.
017532  000461          BR      60$             ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
                        ;*
                        ; WE ARE NOW SYNCHRONIZED WITH THE LTC.
                        ; SET UP FOR THE CALIBRATION LOOP.
                        ;-
017534  012704  002340  6$:     MOV     #TIMER1,R4      ;WILL TEST TIMER1 IN THE LOOP BELOW.
```

```
2482 017540  005001                    CLR     R1              ;CLEAR THE OUTER LOOP COUNTER.
2483 017542  005002                    CLR     R2              ;INDICATE TO CHECK ALL BITS OF TIMER1.
2484 017544  005003                    CLR     R3              ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2485 017546  012714  000001            MOV     #1,(R4)         ;LOAD TIMER1 WITH COUNT OF 1.
2486
2487 017552  016705  162574    8$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2488 017556  011400            10$:    MOV     (R4),R0         ;GET THE TIMER1 VALUE.
2489 017560  010067  000120            MOV     R0,64$          ;SAVE WORD (LIKE IN THE REAL LOOP).
2490 017564  040200                    BIC     R2,R0           ;LEAVE ALL THE BITS.
2491 017566  020003                    CMP     R0,R3           ;COMPARE AGAINST ZERO.
2492 017570  000261                    SEC                     ;SET CARRY IN CASE OF SUCCESS.
2493 017572  001406                    BEQ     12$             ;EXIT LOOP IF TIMER1 HAS CLEARED.
2494 017574  005305                    DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2495 017576  001367                    BNE     10$             ;LOOP IF MS NOT UP.
2496 017600  005301                    DEC     R1              ;DECREMENT THE MS TIME COUNT.
2497 017602  001363                    BNE     8$              ;KEEP LOOPING.
2498 017604  004767  000440            JSR     PC,OOPS         ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
2499                                ;+
2500                                ; WE HAVE NOW HAVE LOOP COUNT INFORMATION FOR ONE CLOCK TICK.
2501                                ; WE HAVE NEGATIVE OF NUMBER OF OUTER LOOPS IN R1, EACH IS MSLCNT INNER LOOPS.
2502                                ; WE HAVE THE PORTION OF THE LAST OUTER LOOP NOT EXECUTED, IN R5.
2503                                ; NOW WE CALCULATE THE TOTAL NUMBER OF INNER LOOPS EXECUTED.
2504                                ;-
2505 017610  005401            12$:    NEG     R1              ;GET NUMBER OF OUTER LOOPS.
2506 017612  016702  162534            MOV     MSLCNT,R2       ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2507 017616  010203                    MOV     R2,R3           ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2508 017620  160502                    SUB     R5,R2           ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2509 017622  010204                    MOV     R2,R4           ;   AND ADD TO ACCUMULATOR LSWORD.
2510 017624  005005                    CLR     R5              ;CLEAR ACCUMULATOR MSWORD.
2511 017626  005301            14$:    DEC     R1              ;CHECK R1 FOR 0 CONDITION
2512 017630  100403                    BMI     16$             ; SKIP MULTIPLICATION IF ZERO
2513 017632  060304                    ADD     R3,R4           ;MULTIPLY NUMBER OF INNER
2514 017634  005505                    ADC     R5              ; LOOPS PER OUTER LOOP BY
2515 017636  000773                    BR      14$             ;NUMBER OF OUTER LOOPS PERFORMED.
2516                                ;+
2517                                ; DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER LTC TICK.
2518                                ;-
2519 017640  016701  162504    16$:    MOV     MSTICK,R1       ;# OF MS PER LTC TICK IS DIVISOR.
2520 017644  010403                    MOV     R4,R3           ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2521 017646  010502                    MOV     R5,R2           ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2522 017650  004767  003140            JSR     PC,UNSDIV       ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
2523 017654  103402                    BCS     18$             ;BYPASS OOPS IF WE'RE OK.
2524 017656  004767  000366            JSR     PC,OOPS         ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2525 017662  010167  162464    18$:    MOV     R1,MSLCNT       ;SET NEW VALUE FOR MS LOOP COUNT.
2526 017666  005167  000010            COM     62$             ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2527 017672  001277                    BNE     2$              ;BRANCH IF ONLY ONE ITERATION DONE.
2528 017674  000261                    SEC                     ;SET THE SUCCESS FLAG FOR EXIT.
2529
2530 017676                    60$:    PASS                    ;RESTORE GPRS.
     017676  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
2531 017700  000207                    RTS     PC              ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
2532
2533 017702  000000            62$:    .WORD   0               ;2ND CALIBRATION ITERATION FLAGS.
2534 017704  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.
                                ;*****************************************************************************
                                
017706                          CKTRAP:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
017706  004567  164060                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017712  005067  162376                 CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAGS.
017716  011011                         MOV     (R0),(R1)       ;PERFORM THE MOVE IN QUESTION.
017720  005767  162370         ADRPTR:: TST    TP4FLG          ;CHECK FOR OCCURENCE OF TRAP.
017724  000261                         SEC                     ;INDICATE SUCCESS.
017726  001401                         BEQ     60$             ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
017730  000241                         CLC                     ;INDICATE FAILURE.
017732                          60$:    PASS                    ;RESTORE GPRS.
017732  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
017734  000207                         RTS     PC
```

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

017736                  CLNRST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017736  004567  164030                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; RESET THE DUT.
                        ;  THIS ROUTINE REPORTS ERRORS WITH NUMBERS FROM ERRNBR THRU ERRNBR+2.
                        ;-
017742  004767  001344          JSR     PC,RESETT       ;RESET THE DUT TO A KNOWN STATE.
017746  103002                  BCC     60$             ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
                        ;+
                        ; PURGE THE FIFO OF ERROR CODES, SAVE ANY BMP CODES FOUND.
                        ;-
017750  004767  000522          JSR     PC,PUFIFO       ;PURGE THE FIFO.

017754                  60$:                            ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
017754                          PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
017754  004736                          JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
                                                        ;CARRY BIT:IF CLEAR, THEN ABORT THE TEST.
017756  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.
                                ;-- ************************************************************************************

017760                          CLR16W:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017760  004567  164006                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017764  012701  000020                MOV     #16.,R1                 ;SET THE LOOP COUNTER TO 16.
017770  005020                  2$:   CLR     (R0)+                   ;CLEAR A WORD OF MEMORY.
017772  005301                        DEC     R1                      ;COUNT THIS LOOP.
017774  001375                        BNE     2$                      ;LOOP IF NOT 16 WORD CLEARED.
017776                          60$:  PASS                            ;RESTORE GPRS.
017776  004736                                JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020000  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.
                                ;-- ************************************************************************************
                                CNTERR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020002  004567  163764                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;*
                                ; COUNT THE ERROR ON THE COUNTER FOR THE SPECIFIED LINE.
                                ;-
020006  006305                          ASL     R5                      ;FORM WORD OFFSET FROM LINE NUMBER.
020010  016501  002460                  MOV     ERCNTB(R5),R1           ;GET THE PRESENT ERROR COUNT FOR THIS LINE.
020014  005201                          INC     R1                      ;COUNT ERROR.
020016  103402                          BCS     2$                      ;OVERFLOW?  YES, DON'T UPDATE COUNTER IN TABLE.
020020  010165  002460                  MOV     R1,ERCNTB(R5)           ;UPDATE ERROR COUNTER TABLE ENTRY.
020024  005767  162172          2$:     TST     NDERPT
020030  001411                          BEQ     60$                     ;SUMMARYS DISABLED?  YES, EXIT WITH CARRY 0.
020032  020167  162164                  CMP     R1,NDERPT               ;NO.  CHECK FOR ENOUGH ERRORS FOR SUMMARY USE.
020036  101002                          BHI     4$                      ;ENOUGH ERRORS TO USE SUMMARY?  YES, GO HANDLE.
020040  000241                          CLC                             ;INDICATE NOT TO USE SUMMARY REPORT YET.
020042  000404                          BR      60$                     ;EXIT WITH CARRY 0.
020044  056567  002404  162404  4$:     BIS     BITTBL(R5),ERSMRF               ;SET THE ERROR SUMMARY FLAG FOR LINE.
020052  000261                          SEC                             ;INDICATE TO USE SUMMARY REPORT.
020054                          60$:    PASS                            ;RESTORE GPRS.
020054  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020056  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.
                                ;*****************************************************************************

020060                          DELAY:: SAVE                            ;SAVE CONTENTS OF GPRS RO THRU R5.
020060  004567  163706                          JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
020064  010401                  MOV     R4,R1                           ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
020066  012702  177777          MOV     #-1,R2                          ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
020072  005003                  CLR     R3                              ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
020074  012704  020116          MOV     #62$,R4                         ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
020100  004767  000130          JSR     PC,MSLOOP                       ;DELAY THE REQUESTED # OF MS.
020104  103002                  BCC     60$                             ;EXIT ROUTINE IF WE TIMED-OUT.]
020106  004767  000136          JSR     PC,OOPS                         ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
020112                          60$:    PASS                            ;RESTORE GPRS.
020112  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
020114  000207                  RTS     PC

020116  177777                  62$:    .WORD   -1                      ;DUMMY, NON-ZERO WORD.
```

J6


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

020120                  MSLGET:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020120  004567  163646                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; SET UP MASK FOR REMOVING UNUSED BITS IN THE TEST WORD, AND CLEAR UNUSED
                        ; BITS IN THE DESIRED STATE WORD TO ALLOW DIRECT COMPARISON.
                        ;-
020124  005102                  COM     R2              ;GET MASK OF UNUSED BITS.
020126  040203                  BIC     R2,R3           ;MASK OUT UNUSED BITS IN DESIRED STATE WORD.
                        ;*
                        ; HANDLE THE TEST AND EXIT IF WE HAVE A 0 TIME-OUT VALUE.
                        ;-
020130  005701                  TST     R1              ;TEST THE TIME-OUT VALUE FOR ZERO.
020132  001011                  BNE     2$              ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
020134  011400                  MOV     (R4),R0         ;GET THE WORD TO TEST BEFORE EXITING.
020136  010067  000070          MOV     R0,62$          ;SAVE VALUE SO WE CAN RETURN IT.
020142  040200                  BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
020144  020003                  CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
020146  000261                  SEC                     ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
020150  001420                  BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
```

```
2768 020152  000241                    CLC                     ;INDICATE FAILURE (TIME-OUT).
2769 020154  000416                    BR      6$              ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
2770                                   ;+
2771                                   ; NON-ZERO TIME-OUT VALUE.  LOOP, WAITING FOR CONDITION OR TIME-OUT.
2772                                   ;-
2773 020156  016705  162170            2$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2774 020162  011400                    4$:     MOV     (R4),R0         ;GET THE WORD TO TEST.
2775 020164  010067  000042                    MOV     R0,62$          ;SAVE WORD IN CASE THIS IS THE LAST.
2776 020170  040200                            BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
2777 020172  020003                            CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
2778 020174  000261                            SEC                     ;SET CARRY IN CASE OF SUCCESS.
2779 020176  001405                            BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
2780 020200  005305                            DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2781 020202  001367                            BNE     4$              ;LOOP IF MS NOT UP.
2782 020204  005301                            DEC     R1              ;DECREMENT THE MS TIME COUNT.
2783 020206  001363                            BNE     2$              ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
2784 020210  000241                            CLC                     ;CLEAR CARRY, WE TIMED-OUT.
2785                                   ;+
2786                                   ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME-OUT VALUE).
2787                                   ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD.  EXIT ROUTINE.
2788                                   ;-
2789 020212  016700  000014            6$:     MOV     62$,R0          ;PASS OUT THE LAST READ WORD.
2790 020216                            60$:    PASS    R0,R1           ;RESTORE GPRS, EXCEPT THE FOLLOWING:
     020216  010066  000002                                    MOV     R0,R0SLOT(SP)           ;PUT R0 IN STACK SLOT.
     020222  010166  000004                                    MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
     020226  004736                                            JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2791                                                                 ;R0 - LAST READ WORD CHECKED FOR CONDITION.
2792                                                                 ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
2793 020230  000207                            RTS     PC              ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
2794                                   ;+
2795                                   ; LOCAL STORAGE.
2796                                   ;-
2797 020232  000000                    62$:    .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.
                                ;*************************************************************************

020234                          MSLOOP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020234  004567  163532                          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.
                                ;-
020240  004767  177654                  JSR     PC,MSLGET               ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.

020244                          60$:    PASS                            ;RESTORE GPRS,
020244  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020246  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.
                                ;-- ************************************************************************

020250                          OOPS::  SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
020250  004567  163516                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                    ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
020254                              ERRSF   101,EM0101
020254  104454                                                          TRAP    C$ERSF
020256  000145                                                          .WORD   101
020260  020314                                                          .WORD   EM0101
020262  000000                                                          .WORD   0
                                    ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
020264                              PRINTF  #EM0102
020264  012746  020400                                                  MOV     #EM0102,-(SP)
020270  012746  000001                                                  MOV     #1,-(SP)
020274  010600                                                          MOV     SP,R0
020276  104417                                                          TRAP    C$PNTF
020300  062706  000004                                                  ADD     #4,SP
020304                          2$:     BREAK                   ;LOOK FOR OPERATOR CONTROL-C INPUT.
020304  104422                                                          TRAP    C$BRK
020306  000776                  BR      2$                      ;INFINITE LOOP.
020310                          60$:    PASS                    ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
020310  004736                                  JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
020312  000207                  RTS     PC                      ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.

020314      110     117     123 EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
020317      124     040     103
020322      117     115     120
020325      125     124     105
020330      122     040     110
020333      101     122     104
020336      127     101     122
020341      105     040     117
020344      122     040     123
020347      117     106     124
020352      127     101     122
020355      105     040     102
020360      125     107     040
020363      105     116     103
020366      117     125     116
020371      124     105     122
```

```
        020374      105      104      056
        020377      000
  2873  020400      045      116      045  EM0102:: .ASCIZ /%N%APROGRAM HUNG, WAITING FOR A CONTROL-C. <**************%N%N/
        020403      101      120      122
        020406      117      107      122
        020411      101      115      040
        020414      110      125      116
        020417      107      054      040
        020422      127      101      111
        020425      124      111      116
        020430      107      040      106
        020433      117      122      040
        020436      101      040      103
        020441      117      116      124
        020444      122      117      114
        020447      055      103      056
        020452      040      074      052
        020455      052      052      052
        020460      052      052      052
        020463      052      052      052
        020466      052      052      052
        020471      045      116      045
        020474      116      000
  2874                                      .EVEN
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - PUFIFO -
                                ;*************************************************************************
                                ;*      - PURGE THE FIFO
                                ;*      THIS ROUTINE TRIES TO REMOVE ALL THE CHARACTERS FROM THE FIFO.
                                ;*      ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE BMP CODE QUEUE.
                                ;*
                                ;* INPUTS:      RBUFA- CONTAINS THE  ADDRESS OF THE RECEIVER.
                                ;*
                                ;*
                                ;* OUTPUTS:     CARRY BIT - INDICATES THE STATE OF THE FIFO, SET:= PURGED.
                                ;*              BMPCQ - THE CONTENTS OF THE BMP CODE QUEUE MAY BE UPDATED.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,PUFIFO
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
                                ;*************************************************************************

020476                          PUFIFO::SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
020476  004567  163270                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020502  012701  001000                MOV     #512.,R1                ;SET MAXIMUM TRY COUNT OF 512.
020506  016704  161526                MOV     RBUFA,R4                ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.

020512  011402                  2$:     MOV     (R4),R2                 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
020514  100016                        BPL     6$                      ;EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.
                                ;*
                                ; CHECK IF THE READ CHARACTER IS ACTUALLY A BMP CODE.
                                ;  IF IT IS, THEN SAVE IT ON THE BMP CODE QUEUE TO BE REPORTED LATER.
                                ;-
020516  012700  070000                MOV     #70000,R0               ;GENERATE A BIT MAP OF CHAR ERROR BITS
020522  040200                        BIC     R2,R0                   ; WHICH ARE NOT SET FOR CHAR.
020524  001006                      BNE     4$                      ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
                                  ;*
                                  ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
                                  ;-
020526  012700  000301                MOV     #301,R0                 ; CHECK IF BMP.
020532  040200                        BIC     R2,R0                   ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
020534  001002                      BNE     4$                      ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
020536  004767  001470                JSR     PC,SAVBMP               ;SAVE BMP CODE ON THE QUEUE.

020542  005301                  4$:     DEC     R1                      ;DECREMENT THE TRY COUNT.
020544  001362                      BNE     2$                      ;LOOP TO TRY AGAIN.
020546  000241                       CLC                            ;CLEAR CARRY,TO INDICATE FIFO NOT PURGED.
020550  000401                      BR      60$                     ;EXIT WITH CARRY CLEAR.
020552  000261                  6$:     SEC                             ;SET CARRY, TO INDICATE FIFO PURGED.

020554                          60$:    PASS                            ;RESTORE GPRS,
020554  004736                                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                        ;CARRY BIT, SET INDICATES FIFO PURGED.
020556  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.
                        ;-- *************************************************************************

020560                  RDPDR:: SAVE                            ;SAVE CONTENTS OF GPRS RO THRU R5.
020560  004567  163206          JSR     R5,PREGOS               ;CALL REGISTER SAVE SUBRT.
020564  012767  016352  163176        MOV     #ER1601,ERRBLK  ;SET UP THE ADDRESS OF THE ERROR REPORT RTN.
                        ;+
                        ; DETERMINE WHETHER REGISTER DATA SHOULD BE INVERTED FROM DATA PATTERN.
                        ;-
020572  005704                  TST     R4                      ;CHECK THE OPERAND TYPE INDICATOR.
020574  100001                BPL     2$                      ;BIC WRITE PERFORMED?  NO, USE STANDARD DATA.
020576  005102                  COM     R2                      ;YES, INVERT THE DATA PATTERN.
                        ;+
                        ; SET UP OUTER LOOP.
                        ;-
020600  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.
                        ;-
```

```
2982 020602  010267  000222          4$:     MOV     R2,70$          ;SAVE THE OUTER LOOP DATA PATTERN.
2983 020606  010577  161424                  MOV     R5,@CSRA        ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
2984 020612  010500                          MOV     R5,R0
2985 020614  006300                          ASL     R0
2986 020616  036067  002404  161400          BIT     BITTBL(R0),ACTLNS
2987 020624  001467                        BEQ     16$             ;IS THE LINE ACTIVE?  NO, SKIP THE LINE.
2988 020626  012703  000004                  MOV     #LPRO,R3        ;YES, INITIALIZE REGISTER OFFSET FOR LPR.
2989                                      ;+
2990                                      ;       THE INNER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP READS AND COMPARES
2991                                      ;       DATA FROM A DEVICE REGISTER.
2992                                      ;-
2993 020632  010204                  6$:     MOV     R2,R4           ;SAVE THE INNER LOOP DATA PATTERN.
2994 020634  046302  002364                  BIC     UNBTTB(R3),R2   ;REMOVE UNUSED BITS FROM EXPECTED DATA.
2995 020640  016300  002236                  MOV     DRADRT(R3),R0
2996 020644  005766  000010                  TST     R3SLOT(SP)      ;CHECK THE ACCESS TYPE INDICATOR.
2997 020650  001002                        BNE     8$              ;BYTE ACCESS?  YES, GO PERFORM BYTE READ.
2998 020652  011001                         MOV     (R0),R1         ;NO, PERFORM WORD READ OF DEVICE REGISTER.
2999 020654  000416                        BR      12$
3000 020656  100410                    8$:     BMI     10$             ;LOW BYTE ACCESS?  YES, GO DO LOW BYTE READ.
3001 020660  005200                          INC     R0              ;HIGH BYTE ACCESS.  FORM HIGH BYTE ADDRESS.
3002 020662  111001                          MOVB    (R0),R1         ;READ THE HI BYTE OF THE DUT REGISTER.
3003 020664  000301                          SWAB    R1              ;PUT HI BYTE BACK INTO THE HI BYTE.
3004 020666  042701  000377                  BIC     #377,R1         ;REMOVE THE UNUSED BYTE IN ACTUAL DATA.
3005 020672  042702  000377                  BIC     #377,R2         ;REMOVE THE UNUSED BYTE IN EXPECTED DATA.
3006 020676  000405                        BR      12$
3007 020700  111001                   10$:     MOVB    (R0),R1         ;READ THE LOW BYTE OF THE DUT REGISTER.
3008 020702  042701  177400                  BIC     #177400,R1      ;REMOVE THE UNUSED BYTE.
3009 020706  042702  177400                  BIC     #177400,R2      ;FORM EXPECTED LOW BYTE FOR COMAPARISON.
3010
3011 020712  046301  002364             12$:     BIC     UNBTTB(R3),R1   ;REMOVE UNUSED BITS FROM ACTUAL DATA.
3012 020716  020102                          CMP     R1,R2           ;COMPARE ACTUAL AND EXPECTED DATA.
3013 020720  001414                        BEQ     14$             ;ACTUAL = EXPECTED?  YES, SKIP ERROR.
3014 020722  004767  177054                  JSR     PC,CNTERR       ;NO, COUNT THE ERROR, CHECK FOR ERROR SUMMARY.
3015 020726  103411                        BCS     14$             ;USE ERROR SUMMARY?  YES, SKIP ERROR.
3016                                  ;NO, REPORT "BAD BIT(S) IN DEVICE XXXXX REGISTER FOR LINE NN (D)."
3017 020730                                   ERROR
     020730  104460                                                                          TRAP    C$ERROR
3018                                      ;+
3019                                      ;EXIT THIS ROUTINE AND SET THE "EXIT ON ERROR" FLAG, IF EXTENDED ERROR
3020                                      ;REPORTING HAS NOT BEEN REQUESTED.
3021                                      ;-
3022 020732  032767  000100  161260          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3023 020740  001004                        BNE     14$             ;BRANCH IF IT HAS.
3024 020742  012767  000001  161330          MOV     #1,EXOERR       ;SET THE EXIT ON ERROR FLAG.
3025 020750  000425                        BR      60$             ;EXIT THE ROUTINE.
3026
3027 020752  010402                   14$:     MOV     R4,R2           ;RESTORE THE INNER LOOP DATA PATTERN.
3028 020754  004767  000444                  JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
3029 020760  062703  000002                  ADD     #2,R3           ;SET REGISTER OFFSET TO THE NEXT REGISTER.
3030 020764  020327  000006                  CMP     R3,#FSLSO       ;CHECK THAT THIS IS NOT THE FIFOSIZE/DATA REG.
3031 020770  001002                        BNE     15$             ;AVOID ALTERING THE OFFSET IF IT ISN'T
3032 020772  062703  000002                  ADD     #2,R3           ;POINT AT THE NEXT REGISTER.
3033 020776  020327  000016             15$:     CMP     R3,#TXBFCO      ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
3034 021002  003713                         BLE     6$              ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
3035                                      ;+
3036                                      ; BACK INTO THE OUTER LOOP.  NOW SET UP FOR NEXT LINE.  LOOP IF NOT DONE.
3037                                      ;-
```

```
3038 021004  016702  000020          16$:    MOV     70$,R2          ;SET UP TO ROTATE THE DATA PATTERN.
3039 021010  004767  000410                  JSR     PC,ROLDAP       ;ROTATE THE DATA PATTERN.
3040 021014  005205                          INC     R5              ;COUNT THIS LINE
3041 021016  020527  000020                  CMP     R5,@NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
3042 021022  002667                          BLT     4$              ;LOOP IF SOME LINES NOT DONE.
3043
3044 021024                          60$:    PASS                    ;RESTORE GPRS.
     021024  004736                                  JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
3045 021026  000207                  RTS     PC
3046
3047 021030  000000                  70$:    .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,ROLDAP,SWAPO,WDPDR
                                ;-- ************************************************************************

021032                          REGTST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021032  004567  162734                          JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
                                ;+
                                ; SET UP THE GPRS FOR THE WRITTING OF THE DATA PATTERN.
                                ;-
021036  012705  000020                  MOV     #16.,R5         ;SET UP LOOP COUNTER TO COUNT 16 ITERATIONS.
021042  012702  167410                  MOV     #167410,R2      ;INITIALIZE THE DATA PATTERN.
021046  032704  000001                  BIT     #BIT0,R4        ;TEST FOR R/W ACCESS.
021052  001001                          BNE     2$              ;R/M/W ACCESS?  YES, R4 IS ALL SET UP.
021054  005004                          CLR     R4              ;NO, INDICATE R/W ACCESS.
021056                          2$:
                                ;+
                                ; SET UP THE GPRS FOR THE CLEARING OR SETTING OF ALL THE USED BITS.
                                ;-
021056  010400                          MOV     R4,R0           ;PASS OPERATION TYPE INDICATOR AROUND SWAPO.
021060  004767  001272                  JSR     PC,SWAPO        ;GET ALTERNATE GPR SET IN R1 THRU R5.
021064  016701  162674                  MOV     ERRNBR,R1       ;SAVE THE INITIAL ERROR NUMBER.
021070  010004                          MOV     R0,R4
021072  005404                          NEG     R4              ;SET UP OP TYPE FOR CLEARING OR SETTING.
021074  005002                          CLR     R2              ;SET UP CLEAR WRITE PATTERN.
021076  026627  000012  000002          CMP     R4SLOT(SP),#2   ;TEST FOR CLEAR THEN MOV TEST SEQUENCE.
021104  001401                          BEQ     4$              ;CLEAR THEN MOV?  YES, LEAVE WRITE PAT CLEAR.
021106  005102                          COM     R2              ;NO, SET ALL BITS OF WRITE PATTERN.
021110  005003                  4$:     CLR     R3              ;INDICATE THAT WORD ACCESSES SHOULD BE USED.
021112  005000                          CLR     R0              ;SET ALTERNATE BYTE EXPECTED DATA PAT TO CLEAR.
021114  026627  000012  177776          CMP     R4SLOT(SP),#-2  ;TEST FOR SET THEN MOV TEST SEQUENCE.
021122  001001                          BNE     6$              ;SET THEN MOV?  YES, LEAVE ALT BYTE PAT CLEAR.
021124  005100                          COM     R0              ;NO, SET ALT BYTE EXPECTED DATA PAT TO ALL 1'S.
021126  004767  001224          6$:     JSR     PC,SWAPO        ;RESTORE SWAPPED GPR VALUES TO R1 THRU R5.
                                ;+
                                ; START OF DATA PATTERN LOOP.
```

```
                                        ;-
021132                                  8$:
                                            ;+
                                            ; SET OR CLEAR ALL THE USED BITS OF THE DEVICE REGISTERS FOR ALL LINES.
                                            ; VERIFY THAT ALL THE BITS WERE SET OR CLEARED CORRECTLY.
                                            ;-
021132  004767  001220                          JSR     PC,SWAPO        ;GET ALTERNATE GPRS FOR SETTING INTIAL STATES.
021136  004767  002062                          JSR     PC,WDPDR        ;GO CLEAR ALL USED REGISTER BITS, ALL LINES.
021142  010167  162616                           MOV    R1,ERRNBR       ;SET UP ERROR NUMBER TO INITIAL ERRNBR.
021146  004767  177406                          JSR     PC,RDPDR        ;VERIFY ALL USED REGISTER BITS, ALL LINES.
                                        ;+
                                        ;EXIT THIS ROUTINE IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING
                                        ;HAS BEEN REQUESTED.  I.E.  EXOERR IS NON-ZERO.
                                        ;-
021152  005767  161122                           TST    EXOERR          ;HAS AN ERROR BEEN FOUND ?
021156  001035                                  BNE     60$             ;EXIT THIS ROUTINE IF IT HAS.

021160  004767  001172                          JSR     PC,SWAPO        ;RESTORE MAIN GPRS CONTENTS.
                                            ;+
                                            ; WRITE DATA PATTERNS, ALL LOWER BYTE USED BITS, ALL REGISTERS, ALL LINES.
                                            ; VERIFY THAT THE DATA PATTERN WAS WRITTEN CORRECTLY.
                                            ;-
021164  004767  002034                          JSR     PC,WDPDR        ;WRITE DATA PATTERN TO DEVICE REGISTERS.
021170  005267  162570                           INC    ERRNBR          ;SET ERROR NUMBER TO INITIAL+1.
021174  004767  177360                          JSR     PC,RDPDR        ;VERIFY DATA PATTERN IN ALTERRED BYTE(S).
                                        ;+
                                        ;EXIT THIS ROUTINE IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING
                                        ;HAS BEEN REQUESTED.
                                        ;-
021200  005767  161074                           TST    EXOERR          ;HAS AN ERROR BEEN FOUND ?
021204  001022                                  BNE     60$             ;EXIT THIS ROUTINE IF IT HAS.

021206  005703                                   TST    R3              ;CHECK THE BYTE INDICATOR.
021210  001414                                  BEQ     10$             ;WORD ACCESS?  YES, SKIP SECOND BYTE CHECK.
                                            ;+
                                            ; CHECK THAT THE ALTERNATE (UNMODIFIED) BYTE IS CLEAR OR SET AS EXPECTED.
                                            ;-
021212  010201                                   MOV    R2,R1           ;SAVE THE DATA PATTERN.
021214  010002                                   MOV    R0,R2           ;GET THE ALTERNATE BYTE EXPECTED DATA.
021216  005403                                   NEG    R3              ;INDICATE THAT OTHER BYTE IS TO BE CHECKED.
021220  005267  162540                           INC    ERRNBR          ;SET ERROR NUMBER TO INITIAL+2.
021224  004767  177330                          JSR     PC,RDPDR        ;VERIFY DATA PATS IN OTHER BYTES OF REGISTERS.
                                        ;+
                                        ;EXIT THIS ROUTINE IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING
                                        ;HAS BEEN REQUESTED.
                                        ;-
021230  005767  161044                           TST    EXOERR          ;HAS AN ERROR BEEN FOUND ?
021234  001006                                  BNE     60$             ;EXIT THIS RO  INE IF IT HAS.

021236  005403                                   NEG    R3              ;RESTORE BYT    DICATOR.
021240  010102                                   MOV    R1,R2           ;RESTORE DA    ATTERN.
                                            ;+
                                            ; PEPARE THE NEXT DATA PATTERN AND LOOP IF NOT DONE.
                                            ;-
021242  004767  000156                  10$:    JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
021246  005305                                   DEC    R5              ;COUNT THIS ITERATION OF THE LOOP.
021250  003330                                  BGT     8$              ;ALL PATTERNS DONE?  NO, LOOP.
```

```
 3162
 3163 021252  016767  161166  162504  60$:    MOV     GPRS0B,ERRNBR   ;YES, RESTORE ERROR NUMBER AND EXIT.
 3164 021260                                  PASS                    ;GET THE ERROR NUMBR FROM GPR SWAP STORAGE.
      021260  004736                                      JSR         ;RESTORE GPRS.
 3165 021262  000207                          RTS     PC              PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
```

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

021264                  REPSMR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021264  004567  162502                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021270  005767  161162          TST     ERSMRF                  ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
021274  001404                  BEQ     60$                     ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
                        ;*
                        ; WE HAVE SOME ERROR SUMMARIES TO REPORT.
                        ;-
021276  012767  016656  162464          MOV     #ER9004,ERRBLK  ;SELECT ERROR REPORTING ROUTINE.
                        ;*
                        ; REPORT
                        ;    "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
                        ;-
021304                      ERROR
021304  104460                                                                 TRAP    C$ERROR

021306                  60$:    PASS                            ;RESTORE GPRS.
021306  004736                          JSR     PC,@(SP)+                   ;RETURN TO PREG05 SUBRT.
021310  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.
                        ;*****************************************************************************

021312                  RESETT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
021312  004567  162454          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021316  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.
                        ;-
021322  016704  160710          MOV     CSRA,R4         ;GET THE ADDRESS OF THE DUT'S CSR.
021326  030214                  BIT     R2,(R4)         ;CHECK STATE OF MASTER RESET BIT.
021330  001406                  BEQ     2$              ;DON'T DELAY IF MR IS ALREADY CLEAR.
021332  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
021334  012701  011610          MOV     #5000.,R1       ;PASS TIME-OUT VALUE OF 5 SECONDS.
021340  004767  176554          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
021344  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 5 SECS, JUST IN CASE THE SELF-TEST EXECUTES.
                        ;-
021346  010277  160664  2$:      MOV     R2,@CSRA        ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
021352  004767  000722          JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
                          ;*
                          ;   SET SELF-TEST TIME-OUT OF 5 SECONDS, AND WAIT FOR M.R TO CLEAR.
                          ;   IF TIME-OUT OCCURS, THEN REPORT THE FATAL ERROR AND PASS-OUT THE ABORT
                          ;   TEST INDICATOR.
                          ;-
021356  005003                    CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
021360  012701  011610          MOV     #5000.,R1       ;PASS TIME-OUT VALUE OF 5 SECONDS.
021364  004767  176530          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
021370  103410                  BCS     6$              ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
```

```
3265                                    ;+
3266                                    ; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET,TEST ABORTED".
3267                                    ; INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
3268                                    ;-
3269 021372  012701  012171            4$:     MOV     #EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
3270 021376  012767  016462  162364            MOV     #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
3271                                            ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
3272                                            ; "TEST ABORTED"
3273 021404                                    ERROR                   ;               >>>>> ERROR <<<<<
     021404  104460                                                                            TRAP    C$ERROR
3274 021406  000241                            CLC                     ;INDICATE TEST IS TO BE ABORTED.
3275 021410  000403                            BR      60$             ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
3276                                    ;+
3277                                    ; CLEAR TX AND RX INTERRUPT ENABLE STATUS FLAGS IN IESTAT.
3278                                    ; EXIT WITH CONTINUE TEST INDICATOR SET (IE,CARRY SET).
3279                                    ;-
3280 021412  005067  160666            6$:      CLR    IESTAT          ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
3281 021416  000261                             SEC                    ;INDICATE SUCCESS, CONTINUE TEST.
3282
3283 021420                            60$:    PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
     021420  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3284                                                                    ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
3285 021422  000207                            RTS     PC
3286
```

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

021424                          ROLDAP::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
021424  004567  162342                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021430  005702                          TST     R2              ;CHECK MSB, AND CLEAR CARRY.
021432  100001                        BPL     2$              ;BRANCH IF CLEAR.
021434  000261                          SEC                     ;SET CARRY IF MSB SET
021436  006102                  2$:     ROL     R2              ;ROTATE DATA PATTERN LEFT
021440                          60$:    PASS    R2              ;RESTORE GPRS,EXCEPT
021440  010266  000006                        MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
021444  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;R2 - CONTAINS THE ROTATED DATA PATTERN
021446  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.
                        ;-- ************************************************************************
                        RSTRPT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021450  004567  162316                  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.
                        ;-
021454  005003                          CLR     R3              ;CLEAR THE CODE COUNTER.
021456  016705  162302                  MOV     ERRNBR,R5       ;SAVE ERRNBR FOR RESTORATION LATER.
021462  017702  160552          2$:     MOV     @RBUFA,R2       ;READ A CHAR FROM THE DUT FIFO.
021466  100422                      BMI     4$                  ;SKIP ERROR IF DATA.VALID SET FOR CHAR.
                            ;+
                            ; WE EXPECT A SELFTEST CODE, BUT THIS FIFO SLOT IS EMPTY.
                            ;-
021470  010567  162270              MOV     R5,ERRNBR           ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
021474  012701  015123              MOV     #EM9018,R1          ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
021500  012767  016772  162262      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."
                            ;-
021506                      ERROR                               ;               >>>>> ERROR <<<<<.
021506  104460                                                                    TRAP    C$ERROR
                            ;+
                            ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
                            ;-
021510  032767  000100  160502          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
021516  001003                      BNE     3$                  ;AVOID SET THE FLAG IF IT HAS.
021520  012767  000001  160552          MOV     #1,EXOERR       ;SET THE EXIT ON ERROR FLAG
```

```
3371                                        ;+
3372                                        ; INIDICATE "SUCCESS" (BECAUSE FIFO IS PURGED), AND EXIT THIS ROUTINE.
3373                                        ;-
3374 021526  000261              3$:        SEC                     ;SET SUCCESS FLAG.
3375 021530  000167  000406                JMP     60$             ;EXIT ROUTINE.
3376                                        ;+
3377                                        ; DETERMINE IF THIS IS NOT A SELFTEST CODE.
3378                                        ;-
3379 021534  012700  070001      4$:        MOV     #70001,R0       ;GENERATE BIT MAP OF ANY CLEAR ERROR BITS OR
3380 021540  040200                        BIC     R2,R0           ;  BIT 0 WHICH ARE CLEAR.
3381 021542  001042                        BNE     8$              ;GO TO REPORT ERROR IF THIS IS NOT A TEST CODE.
3382                                        ;+
3383                                        ; WE HAVE A TEST CODE (EITHER BMP OR SELFTEST CODE).
3384                                        ; DETERMINE WHAT TYPE OF CODE WE HAVE.
3385                                        ;-
3386 021544  032702  000200                BIT     #BIT7,R2        ;TEST ROM VERSION CODE INDICATOR BIT.
3387 021550  001462                        BEQ     10$             ;SKIP ERRORS IF SELFTEST ROM VERSION CODE.
3388 021552  120227  000203                CMPB    R2,#203         ;CHECK IF SKIP SELF TEST CODE.
3389 021556  001457                        BEQ     10$             ;SKIP ERROR REPORT IF SKIP SELF TEST CODE FOUND
3390 021560  120227  000201                CMPB    R2,#201         ;CHECK IF NULL CODE PRESENT.
3391 021564  001454                        BEQ     10$             ;SKIP ERROR REPORT IF SELF TEST NULL CODE.
3392 021566  012700  000300                MOV     #300,R0         ;TEST CODE TYPE BITS FOR BOTH CODE
3393 021572  040200                        BIC     R2,R0           ;  TYPE BITS SET (INDICATING BMP CODE).
3394 021574  001003                        BNE     6$              ;IF IT IS NOT A BMP CODE GO REPORT ERROR.
3395 021576  004767  000430                JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
3396 021602  000445                        BR      10$             ;GO GET THE NEXT CHARACTER FROM THE FIFO.
3397                                        ;+
3398                                        ; WE HAVE A SELFTEST ERROR CODE.
3399                                        ;-
3400 021604  010567  162154      6$:        MOV     R5,ERRNBR       ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
3401 021610  005267  162150                INC     ERRNBR          ;CALCULATE INITIAL ERROR NUMBER PLUS 1.
3402 021614  012701  015150                MOV     #EM9020,R1      ;PASS ERROR MESSAGE INFO TO ER9008 ROUTINE.
3403 021620  012767  017062  162142        MOV     #ER9008,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3404                                        ;+
3405                                        ; REPORT ERROR WITH NUMBER INITIAL ERRNRB + 1.
3406                                        ;    "UNEXPECTED SELFTEST ERROR CODE FOR LINE NN IN FIFO AFTER RESET:"
3407                                        ;-
3408 021626                                 ERROR                   ;               >>>>> ERROR <<<<<.
     021626  104460                                                                       TRAP    C$ERROR
3409                                        ;+
3410                                        ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3411                                        ;-
3412 021630  032767  000100  160362        BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3413 021636  001027                        BNE     10$             ;AVOID SET THE FLAG IF IT HAS AND GO TO
3414                                                                ;THE END OF THE LOOP.
3415 021640  012767  000001  160432        MOV     #1,EXOERR       ;SET THE "EXIT ON ERROR" FLAG
3416 021646  000534                        BR      50$             ;EXIT THE ROUTINE WITH FAILURE SINCE THE FIFO
3417                                                                ;IS NOT PRUGED.
3418                                        ;+
3419                                        ; WE HAVE A NON-SELFTEST CODE (EITHER BMP CODE OR DATA CHAR).
3420                                        ;-
3421 021650  010567  162110      8$:        MOV     R5,ERRNBR       ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
3422 021654  062767  000002  162102        ADD     #2,ERRNBR       ;CALCULATE INITIAL ERROR NUMBER PLUS 2.
3423 021662  012701  015133                MOV     #EM9019,R1      ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
3424 021666  012767  016772  162074        MOV     #ER9007,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3425                                        ;+
3426                                        ; REPORT ERROR WITH NUMBER INITIAL ERRNRB + 2.
```

```
                                        ;   "NON-SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET."
                                        ;-
021674                                      ERROR                   ;               >>>>> ERROR <<<<<.
021674  104460                                                                                TRAP    C$ERROR
                                        ;+
                                        ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
                                        ;-
021676  032767  000100  160314              BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
021704  001004                            BNE     10$             ;AVOID SET THE FLAG IF IT HAS AND GO TO
                                                                    ;THE END OF THE LOOP.
021706  012767  000001  160364              MOV     #1,EXOERR       ;SET THE "EXIT ON ERROR" FLAG
021714  000511                            BR      50$             ;EXIT THE ROUTINE WITH FAILURE.

                                        ;+
                                        ; END OF LOOP, LOOP IF NOT ALL CHARS HAVE BEEN READ FROM THE FIFO.
                                        ;-
021716  005203                      10$:     INC     R3              ;SET CODE COUNTER FOR NEXT ITERATION OF LOOP.
021720  020327  000010                      CMP     R3,#8.          ;TEST FOR ALL CODES READ.
021724  002656                            BLT     2$              ;LOOP IF NOT CHARS READ FROM FIFO.
                                        ;+
                                        ; PURGE THE FIFO UNTIL DATA.VALID IS CLEAR OR UNTIL TOO MANY CHARS ARE READ.
                                        ;-
021726  012704  000022                      MOV     #18.,R4         ;INITIALIZE THE CHARACTER COUNTER.
021732  010567  162026                      MOV     R5,ERRNBR       ;GET INITIAL VALUE OF THE ERROR NUMBER.
021736  062767  000003  162020              ADD     #3,ERRNBR       ;CALCULATE ERROR NUMBER OF NEXT ERROR.
021744  012767  017062  162016              MOV     #ER9008,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
021752  017702  160262                12$:  MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE DUT FIFO.
021756  000261                            SEC                     ;INDICATE SUCCESS IN CASE DATA.VALID IS CLEAR.
021760  100070                            BPL     60$             ;EXIT ROUTINE WITH SUCCESS IF DATA.VALID CLEAR.
                                        ;+
                                        ; WE HAVE A CHARACTER.
                                        ; DETERMINE IF CHARACTER IS A DATA CHARACTER.
                                        ;-
021762  012700  070000                      MOV     #70000,R0       ;TEST BITS 12 THRU 14 OF THE
021766  040200                            BIC     R2,R0           ;  CODE READ FROM THE DUT FIFO.
021770  001403                            BEQ     14$             ;SKIP THIS ERROR IF CODE IS NOT A DATA CHAR.
                                        ;+
                                        ; WE HAVE AN UNEXPECTED DATA CHARACTER: SET UP AND GO TO REPORT ERROR.
                                        ;-
021772  012701  015174                      MOV     #EM9022,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
021776  000423                            BR      22$             ;GO TO REPORT THIS ERROR.
                                        ;+
                                        ; WE HAVE AN UNEXPECTED CODE.
                                        ; DETERMINE IF THE CODE IS A MODEM STATUS CODE.
                                        ;-
022000  032702  000001              14$:    BIT     #BIT0,R2        ;TEST MODEM STATUS INDICATOR BIT OF CODE.
022004  001003                            BNE     16$             ;SKIP THIS ERROR IF NOT MODEM STATUS CODE.
                                        ;+
                                        ; WE HAVE A MODEM STATUS CODE: SET UP AND GO TO REPORT ERROR.
                                        ;-
022006  012701  015213                      MOV     #EM9023,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
022012  000415                            BR      22$             ;GO TO REPORT THIS ERROR.
                                        ;+
                                        ; WE HAVE AN ONBOARD TEST CODE.
                                        ; DETERMINE IF THIS CODE IS A BMP CODE.
                                        ;-
022014  032702  000200              16$:    BIT     #BIT7,R2        ;TEST THE ROM VERSION BIT OF THE CODE.
```

```
3483 022020  001404                        BEQ     18$             ;GOTO SET UP FOR SELFTEST CODE IF ROM VERSION.
3484 022022  012700  000300                 MOV     #300,R0
3485 022026  040200                        BIC     R2,R0           ;TEST THE ERROR TYPE BITS OF THE CODE.
3486 022030  001403                        BEQ     20$             ;SKIP THIS ERROR IF BMP CODE.
3487                                       ;+
3488                                       ; WE HAVE A SELFTEST CODE: SET UP AND GO TO REPORT ERROR.
3489                                       ;-
3490 022032  012701  015235          18$:    MOV     #EM9024,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3491 022036  000403                        BR      22$             ;GO TO REPORT THIS ERROR.
3492                                       ;+
3493                                       ; WE HAVE A BMP CODE: SAVE IT ON THE QUEUE.
3494                                       ;-
3495 022040  004767  000166          20$:    JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
3496 022044  000411                        BR      24$             ;
3497                                       ;+
3498                                       ; REPORT THE ERROR WITH ERROR NUMBER OF INITIAL ERRNBR + 3.
3499                                       ;    "UNEXPECTED XXX XXXX FOR LINE NN IN FIFO AFTER RESET:"
3500                                       ;-
3501 022046                          22$:    ERROR                   ;               >>>>> ERROR <<<<<.
      022046  104460                                                                   TRAP    C$ERROR
3502                                       ;+
3503                                       ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3504                                       ;-
3505 022050  032767  000100  160142        BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3506 022056  001004                        BNE     24$             ;AVOID SETTING THE FLAG IF IT HAS AND GO TO
3507                                                               ;THE END OF THE LOOP.
3508 022060  012767  000001  160212        MOV     #1,EXOERR       ;SET THE "EXIT ON ERROR" FLAG
3509 022066  000424                        BR      50$             ;EXIT THE ROUTINE WITH FAILURE.
3510
3511                                       ;+
3512                                       ; END OF LOOP.
3513                                       ; COUNT THE CHARACTER WE JUST RECEIVED, AND CHECK FOR TOO MANY RECEIVED.
3514                                       ;-
3515 022070  005304                  24$:    DEC     R4              ;COUNT THIS CHARACTER.
3516 022072  001327                        BNE     12$             ;LOOP IF NOT TOO MANY CHARACTERS PURGED.
3517                                       ;+
3518                                       ; WE READ TOO MANY VALID CHARACTERS WHILE TRYING TO PURGE THE FIFO.
3519                                       ; REPORT ERROR AND EXIT WITHOUT SUCCESS.
3520                                       ;    "FIFO WILL NOT PURGE (DATA.VALID STUCK SET), REMAINDER OF TEST SKIPPED."
3521                                       ;-
3522 022074  012701  015012                 MOV     #EM9017,R1      ;SELECT PROPER ERROR MESSAGE.
3523 022100  010567  161660                 MOV     R5,ERRNBR       ;GET INITIAL ERROR NUMBER.
3524 022104  062767  000004  161652         ADD     #4,ERRNBR       ;CALCULATE INITIAL ERRNBR + 4.
3525 022112  012767  016072  161650         MOV     #ER0503,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3526                                       ;PRINT ERROR REPORT.
3527 022120                                ERROR                   ;               >>>>> ERROR <<<<<.
      022120  104460                                                                   TRAP    C$ERROR
3528                                       ;+
3529                                       ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3530                                       ;-
3531 022122  032767  000100  160070         BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3532 022130  001003                        BNE     50$             ;AVOID SETTING THE FLAG IF IT HAS.
3533 022132  012767  000001  160140         MOV     #1,EXOERR       ;SET THE "EXIT ON ERROR" FLAG
3534
3535 022140  000241                  50$:    CLC                     ;CLEAR THE SUCCESS FLAG.
3536
3537 022142                          60$:    PASS                    ;RESTORE GPRS,
```

```
        022142  004736                          JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
3538    022144  000207          RTS     PC              ; CARRY - SUCCESS FLAG (SET IF FIFO IS PURGED).
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - RXIEO -
                                ;** ************************************************************************
                                ;*                        - RECEIVER INTERRUPT DISABLE -
                                ;*      THIS ROUTINE IS USED TO DISABLE RECEIVER INTERRUPTS IN THE DHU11.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      THE RX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
                                ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
                                ;*               ENABLE BITS.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,RXIEO
                                ;*
                                ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*               THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************
                                RXIEO::  MOV    R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
                                         GETPRI -(SP)           ;SAVE PROCESSOR PRIORITY ON STACK.
                                                                         TRAP    C$GPRI
                                                                         MOV     R0,-(SP)
                                         SETPRI #PRI07          ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
                                                                         MOV     #PRI07,R0
                                                                         TRAP    C$SPRI
                                         BIC    #137777,IESTAT  ;CLEAR RX.INT.ENBL BIT IN IESTAT.
                                         MOV    IESTAT,@CSRA    ;DISABLE RX INTERRUPTS.
                                         SETPRI (SP)+           ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
                                                                         MOV     (SP)+,R0
                                                                         TRAP    C$SPRI
                                         MOV    (SP)+,R0        ;RESTORE R0.
                                        RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - RXIE1 -
                                ;++ ************************************************************************
                                ;*                          - RECEIVER INTERRUPT ENABLE -
                                ;*      THIS ROUTINE IS USED TO ENABLE RECEIVER INTERRUPTS IN THE DHU11.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      THE RX.INT.ENBL BIT IS SET IN THE DUT CSR.
                                ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
                                ;*               ENABLE BITS.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,RXIE1
                                ;*
                                ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*               THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

022206  052767  000100  160070  RXIE1::  BIS     #BIT06,IESTAT   ;SET RX.INT.ENBL BIT IN IESTAT.
022214  042767  137677  160062           BIC     #137677,IESTAT  ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
022222  016777  160056  160006           MOV     IESTAT,@CSRA    ;ENABLE RX INTERRUPTS.
022230  000207                           RTS     PC
```

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

022232                          SAVBMP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
022232  004567  161534                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022236  016704  160256                 MOV     BMPCQP,R4               ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
022242  116724  160052                 MOVB    TSTNUM,(R4)+            ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
022246  005204                         INC     R4                      ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
022250  042702  177400                 BIC     #177400,R2              ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
022254  010224                         MOV     R2,(R4)+                ;SAVE THE BMP CODE ON THE QUEUE.
022256  020427  002722                 CMP     R4,#BMPCQE              ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
022262  103402                        BLO     2$                      ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
022264  162704  000004                 SUB     #4,R4                   ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
022270  010467  160224                2$:     MOV     R4,BMPCQP       ;SAVE THE POINTER.

022274                          60$:    PASS                            ;RESTORE GPRS.
022274  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
022276  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.
                        ;-- ************************************************************************

022300                  SKPSTS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
022300  004567  161466                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022304  012704  000012          MOV     #10.,R4                 ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
022310  004767  175544          JSR     PC,DELAY                ;DELAY FOR 10 MILLI-SECONDS.
                        ;+
                        ; WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
                        ;-
022314  012701  000060           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 DHU-11.
022320  012703  052525           MOV     #52525,R3               ;INITIALISE THE SKIP SELF-TEST CODE.
022324  005301          4$:      DEC     R1                      ;SELECT THE NEXT SET OF DEVICE REGISTERS.
022326  016704  157704           MOV     CSRA,R4                 ;GET THE ADDRESS OF THE CSR OF THE DUT.
022332  010124                   MOV     R1,(R4)+                ;SELECT A BANK OF DUT REGISTERS.
022334  010324          6$:      MOV     R3,(R4)+                ;WRITE THE CODE TO A DUT REGISTER.
022336  020467  157712           CMP     R4,TXBFCA               ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
022342  103774                  BLO     6$                      ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
022344  032701  000017           BIT     #17,R1                  ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
022350  001365                  BNE     4$                      ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.

022352                  60$:    PASS                            ;RESTORE GPRS.
022352  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
022354  000207                  RTS     PC
```

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

022356  010046        SWAPO:: MOV     R0,-(SP)        ;SAVE THE CONTENTS OF R0.
                        ;+
                        ; LOAD THE STACK FROM THE GPRS.
                        ;-
022360  010146                MOV     R1,-(SP)        ;SAVE THE CONTENTS OF R1.
022362  010246                MOV     R2,-(SP)        ;SAVE THE CONTENTS OF R2.
022364  010346                MOV     R3,-(SP)        ;SAVE THE CONTENTS OF R3.
022366  010446                MOV     R4,-(SP)        ;SAVE THE CONTENTS OF R4.
022370  010546                MOV     R5,-(SP)        ;SAVE THE CONTENTS OF R5.
                        ;+
                        ; LOAD THE GPRS FROM THE GPR SAVE AREA 0.
                        ;-
022372  012700  002444          MOV     #GPRS0B,R0      ;GET THE BASE ADDRESS OF GPR SAVE AREA 0.
022376  012001                MOV     (R0)+,R1        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 1.
022400  012002                MOV     (R0)+,R2        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 2.
022402  012003                MOV     (R0)+,R3        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 3.
022404  012004                MOV     (R0)+,R4        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 4.
022406  012005                MOV     (R0)+,R5        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 5.
                        ;+
                        ; LOAD THE GPR SAVE AREA 0 FROM THE STACK.
                        ;-
022410  012640                MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 5 WITH SAVED R5.
022412  012640                MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 4 WITH SAVED R4.
022414  012640                MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 3 WITH SAVED R3.
022416  012640                MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 2 WITH SAVED R2.
022420  012640                MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 1 WITH SAVED R1.

022422  012600                MOV     (SP)+,R0        ;RESTORE THE INITIAL VALUE OF R0.

022424  000207              RTS     PC
```

```
3723                                    .SBTTL  GLOBAL SUBROUTINE                - TSABRT -
3724                                    ;++ ************************************************************************
3725                                    ;*                          - TEST ABORT ROUTINE -
3726                                    ;*      THIS SUBROUTINE IS USED WHEN A NON-TEST RELATED ERROR HAS BEEN FOUND
3727                                    ;*      DURING THE EXECUTION OF THE CURRENT TEST.
3728                                    ;*      IT IS USED TO INFORM THE OPERATOR THAT THE CURRENT TEST HAS BEEN
3729                                    ;*      ABORTED.
3730                                    ;*
3731                                    ;* INPUTS:      ERRMSG - CONTAINS THE NAME OF THE CURRENT TEST.
3732                                    ;*               ERRNBR - CONTAINS THE CORRECT ERROR NUMBER.
3733                                    ;*               THE REMAINDER OF THE ERRTBL IS CORRECTLY INITIALISED.
3734                                    ;*
3735                                    ;* OUTPUTS:     MESSAGES ARE REPORTED TO THE OPERATOR.
3736                                    ;*
3737                                    ;* CALLING SEQUENCE:    JSR     PC,TSABRT
3738                                    ;*
3739                                    ;* COMMENTS:
3740                                    ;*
3741                                    ;* SUBORDINATE ROUTINES CALLED: ER1603.
3742                                    ;-- ************************************************************************
3743
3744 022426                             TSABRT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
    022426  004567  161340                              JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3745 022432  012701  022450                      MOV     #2$,R1                  ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
3746 022436  012767  016462  161324              MOV     #ER1603,ERRBLK          ;SET-UP THE ERROR REPORTING ROUTINE.
3747 022444                                     ERROR                           ;               >>>>> ERROR <<<<<.
    022444  104460                                                                                  TRAP    C$ERROR
3748 022446  000432                             BR      60$                     ;
3749 022450     040     116     117  2$:        .ASCIZ / NON-RELATED TEST ERROR FOUND DURING TEST EXECUTION/
     022453     116     055     122
     022456     105     114     101
     022461     124     105     104
     022464     040     124     105
     022467     123     124     040
     022472     105     122     122
     022475     117     122     040
     022500     106     117     125
     022503     116     104     040
     022506     104     125     122
     022511     111     116     107
     022514     040     124     105
     022517     123     124     040
     022522     105     130     105
     022525     103     125     124
     022530     111     117     116
     022533     000
3750                                    .EVEN
3751 022534                             60$:    PASS                            ;RESTORE GPRS.
    022534  004736                                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3752 022536  000207                             RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - TXDSBL -
                                ;++ ************************************************************************************
                                ;*                          - TRANSMITTER DISABLE -
                                ;*      THIS SUBROUTINE IS USED TO DISABLE TRANSMISSION ON SELECTED LINES BY,
                                ;*      CLEARING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
                                ;*
                                ;* INPUTS:      R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO CLEAR TX.ENABLE.
                                ;*              CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
                                ;*              IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
                                ;*              NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
                                ;*              TXAD2A - CONTAINS THE ADDRESS OF THE TBUFFAD2 REGISTER.
                                ;*
                                ;* OUTPUTS:     R5 - BIT'S SET INDICATE THE INITIAL STATES OF ALL TX.ENBLE BITS.
                                ;*              TBUFFAD2 - THE STATE OF THE TX.ENBLE BIT MAY BE ALTERED.
                                ;*              THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TXDSBL
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************************
                                TXDSBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
 022540  004567  161226                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                        MOV     R5,R0                   ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
                                        MOV     #BIT0,R1                ;INITIALIZE THE SELECTED LINE BIT MASK.
                                        MOV     TXAD2A,R2               ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
                                        INC     R2                      ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
                                        MOV     #NUMLNS,R3              ;GET MAXIMUM LINE NUMBER PLUS ONE.
                                        MOV     IESTAT,R4               ;GET THE STATES OF THE INT ENABLE BITS.
                                        CLR     R5                      ;LOG POSSIBLE TX DISABLED ON ALL LINES.
                                ;+
                                ; SELECT EVERY LINE IN TURN, AND LOG THE STATE OF EACH TX.ENABLE BIT.
                                ;-
                                2$:     MOV     R4,@CSRA                ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
                                        TSTB    (R2)                    ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
                                        BPL     4$                      ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
                                        BIS     R1,R5                   ;LOG TX ENABLE BIT SET FOR SELECTED LINE.
                                ;+
                                ; CLEAR TX.ENBLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX DISABLE
                                ; LINE BIT MAP.
                                ;-
                                4$:     BIT     R1,R0                   ;CHECK STATE OF DISABLE LINE BIT MAP.
                                        BEQ     6$                      ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
                                        BICB    #BIT7,(R2)              ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
                                6$:     INC     R4                      ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
                                        ASL     R1                      ;SHIFT BIT MAP FOR NEXT LINE.
                                        DEC     R3                      ;DECREMENT LINE NUMBER.
                                        BNE     2$                      ;LOOP TO CHECK NEXT LINE.

                                60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
 022624  010566  000014                                  MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
 022630  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                 ;R5 - PREVIOUS STATES OF ALL TX.ENABLE BITS.
                                        RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXENBL -
                        ;++ ************************************************************************
                        ;*                        - TRANSMITTER ENABLE -
                        ;*      THIS SUBROUTINE IS USED TO ENABLE TRANSMISSION ON SELECTED LINES BY
                        ;*      SETTING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
                        ;*
                        ;* INPUTS:      R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO SET TX.ENABLE.
                        ;*              CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
                        ;*              IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
                        ;*              NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
                        ;*              TXAD2A - CONTAINS THE ADDRESS OF THE TBUFFAD2 REGISTER.
                        ;*
                        ;* OUTPUTS:     R5 - BIT'S SET INDICATE PREVIOUSLY DISABLED LINES.
                        ;*              TBUFFAD2 - THE STATE OF THE TX.ENBLE BIT MAY BE ALTERED.
                        ;*              THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TXENBL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

022634                  TXENBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
022634  004567  161132                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022640  010500                  MOV     R5,R0                   ;COPY BIT MAP OF LINES TO ENABLE.
022642  012701  000001          MOV     #BIT0,R1                ;INITIALIZE THE SELECTED LINE BIT MASK.
022646  016702  157400          MOV     TXAD2A,R2               ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
022652  005202                  INC     R2                      ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
022654  012703  000020          MOV     #NUMLNS,R3              ;GET MAXIMUM LINE NUMBER.
022660  016704  157420          MOV     IESTAT,R4               ;GET THE STATES OF THE INT ENABLE BITS.
022664  005005                  CLR     R5                      ;CLEAR TX.ENABLE BIT LOG OF DISABLED LINES.
                        ;*
                        ; SELECT EVERY LINE IN TURN,AND LOG ANY TX.ENBLE BIT THAT IS CLEAR.
                        ;-
022666  010477  157344  2$:      MOV     R4,@CSRA                ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
022672  105712                   TSTB    (R2)                    ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
022674  100401                 BMI     4$                      ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
022676  050105                   BIS     R1,R5                   ;LOG TX ENABLE BIT CLEAR FOR SELECTED LINE.
                         ;*
                         ;  SET TX.ENBLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX ENABLE
                         ;  LINE BIT MAP.
                         ;-
022700  030100  4$:       BIT     R1,R0                   ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
022702  001402                 BEQ     6$                      ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
022704  152712  000200           BISB    #BIT7,(R2)              ;ENABLE TRANSMISSION ON SELECTED LINE.
022710  005204  6$:       INC     R4                      ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
022712  006301                   ASL     R1                      ;SHIFT BIT MAP FOR NEXT LINE.
022714  005303                   DEC     R3                      ;DECREMENT LINE NUMBER.
022716  001363                 BNE     2$                      ;LOOP TO CHECK NEXT LINE.

022720                  60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
022720  010566  000014                          MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
022724  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;R5 - LINE BIT MAP CORRESPONDING TO THE
                                                                ; PREVIOUS LINES THAT WERE DISABLED.
022726  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - TXIEO -
                                ;++ ************************************************************************
                                ;*                          - TRANSMITTER INTERRUPT DISABLE -
                                ;*      THIS ROUTINE IS USED TO DISABLE TRANSMITTER INTERRUPTS IN THE DHU11.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      THE TX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
                                ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
                                ;*               ENABLE BITS.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TXIEO
                                ;*
                                ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*               THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************
022730  010046                  TXIEO:: MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
022732                                  GETPRI  -(SP)           ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
022732  104440                                                                  TRAP    C$GPRI
022734  010046                                                                  MOV     R0,-(SP)
022736                                  SETPRI  #PRI07          ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
022736  012700  000340                                                         MOV     #PRI07,R0
022742  104441                                                                  TRAP    C$SPRI
022744  042767  177677  157332          BIC     #177677,IESTAT  ;CLEAR TX.INT.ENBL BIT IN IESTAT.
022752  016777  157326  157256          MOV     IESTAT,@CSRA    ;DISABLE TX INTERRUPTS.
022760                                  SETPRI  (SP)+           ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
022760  012600                                                                  MOV     (SP)+,R0
022762  104441                                                                  TRAP    C$SPRI
022764  012600                          MOV     (SP)+,R0        ;RESTORE R0.
022766  000207                          RTS     PC
```

```
3890                                    .SBTTL  GLOBAL SUBROUTINE                - TXIE1 -
3891                                    ;++ ************************************************************************
3892                                    ;*                        - TRANSMITTER INTERRUPT ENABLE -
3893                                    ;*      THIS ROUTINE IS USED TO ENABLE TRANSMITTER INTERRUPTS IN THE DHU11.
3894                                    ;*
3895                                    ;* INPUTS:       NONE.
3896                                    ;*
3897                                    ;* OUTPUTS:      THE TX.INT.ENBL BIT IS SET IN THE DUT CSR.
3898                                    ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3899                                    ;*               ENABLE BITS.
3900                                    ;*
3901                                    ;* CALLING SEQUENCE:    JSR     PC,TXIE1
3902                                    ;*
3903                                    ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3904                                    ;*               THE DUT CSR ARE DESTROYED.
3905                                    ;*
3906                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
3907                                    ;-- ************************************************************************
3908
3909 022770  052767  040000  157306  TXIE1::  BIS     #BIT14,IESTAT   ;SET TX.INT.ENBL BIT IN IESTAT.
3910 022776  042767  137677  157300          BIC     #137677,IESTAT  ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
3911 023004  016777  157274  157224          MOV     IESTAT,@CSRA    ;ENABLE TX INTERRUPTS.
3912 023012  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.
                                ;-- ************************************************************************

023014                          UNSDIV:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023014  004567  160752                           JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
                                ;*
                                ; CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
                                ;-
023020  010204                          MOV     R2,R4           ;GET MSW OF DIVIDEND FOR SUBTRACT.
023022  160104                          SUB     R1,R4           ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
023024  103403                          BCS     2$              ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
023026  012701  177777                  MOV     #-1,R1          ;SET QUOTIENT TO ALL ONES (177777),
023032  000442                          BR      60$             ;EXIT WITH CARRY CLEAR.
                                ;*
                                ; SET UP COUNTERS AND VARIOUS WORKING GPRS.
                                ;-
023034  005004                  2$:     CLR     R4              ;CLEAR THE LSW OF THE DIVISOR.
023036  000241                          CLC                     ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
023040  006001                          ROR     R1              ; DIVISOR BY
023042  006004                          ROR     R4              ; 2(UNSIGNED)
023044  012700  000020                  MOV     #16.,R0         ;SET UP INITIAL SHIFT COUNT TO 16.
                                ;*
                                ; THE SUBTRACT AND SHIFT LOOP.
                                ;-
023050  010246                  4$:     MOV     R2,-(SP)        ;SAVE MSWORD OF DIVIDEND.
023052  010346                          MOV     R3,-(SP)        ;SAVE LSWORD OF DIVIDEND.
023054  160403                          SUB     R4,R3           ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
023056  005602                          SBC     R2              ;MSWORD DIVIDEND - BORROW .
023060  103402                          BCS     6$              ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
023062  160102                          SUB     R1,R2           ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
023064  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.
                                    ;-
023066  012603                  6$:     MOV     (SP)+,R3        ;RESTORE LSWORD OF DIVIDEND.
023070  012602                          MOV     (SP)+,R2        ;RESTORE MSWORD OF DIVIDEND.
```

```
                        BR      10$             ;GOTO SHIFT 1 INTO THE QUOTIENT.
                    ;+
                    ; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
                    ; COMPLEMENTED LATER).   CARRY IS CLEAR.
                    ;-
                8$: MOV     (SP)+,(SP)+     ;POP THE SAVED DIVIDEND OFF OF THE STACK.
                    ;+
                    ; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.
                    ;-
               10$: ROL     R5              ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
                    CLC                     ;DIVIDE THE
                    ROR     R1              ;  DEVISOR BY
                    ROR     R4              ;  2 (UNSIGNED).
                    DEC     R0              ;COUNT THIS SHIFT AND SUBTRACT.
                    BNE     4$              ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
                    COM     R5              ;GET QUOTIENT FROM INVERTED QUOTIENT.
                ;+
                ; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.
                ;-
                    CLC                     ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
                    ROL     R3              ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
                    BCS     12$             ;IF CARRY FROM SHIFT, ROUND UP.
                    SUB     R4,R3           ;SUBTRACT DIVISOR FROM DIVIDEND.
                    BCS     14$             ;IF BORROW, DON'T ROUND UP.
                ;+
                ; ROUND UP, EXTRA SUBTRACT WENT.
                ;-
               12$: INC     R5              ;INCREMENT THE QUOTIENT BY ONE.
                    BNE     14$             ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
                    DEC     R5              ;DON'T LET ROUNDING CAUSE OVERFLOW.
                ;+
                ; ALL DONE, PASS QUOTIENT AND EXIT.
                ;-
               14$: MOV     R5,R1           ;PASS QUOTIENT BACK IN R1.
                    SEC                     ;INDICATE NO OVERFLOW.

               60$: PASS    R1                      ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
023140 010166 000004                 MOV    R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
023144 004736                        JSR    PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
                                             ;R1 - 16 BIT, UNSIGNED QUOTIENT.
                    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.
                        ;-- ************************************************************************

023150                  WAIBIS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023150  004567  160616                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023154  010204                  MOV     R2,R4                   ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
023156  010102                  MOV     R1,R2
023160  042701  170000          BIC     #170000,R1              ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
023164  042702  007777          BIC     #7777,R2                ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
023170  000302                  SWAB    R2                      ;PUT LINE NUMBER FIELD IN LSBYTE.
023172  006202                  ASR     R2                      ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
023174  006202                  ASR     R2                      ;  POSITION TO USE IT AS A WORD TABLE OFFSET
023176  006202                  ASR     R2                      ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
023200  016202  002404          MOV     BITTBL(R2),R2           ;GET BIT MAP OF LINE TO TEST FROM TABLE.
023204  010203                  MOV     R2,R3                   ;INDICATE THAT THE BIT SHOULD BE SET.
023206  004767  174706          JSR     PC,MSLGET               ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
                                                                ;  CARRY IS CORRECT UPON MSLGET RETURN.
023212  010002                  MOV     R0,R2                   ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
023214                  60$:    PASS    R2                      ;RESTORE GPRS, EXCEPT THE FOLLOWING:
023214  010266  000006                          MOV     R2,R2SLOT(SP)   ;PUT R2 IN STACK SLOT.
023220  004736                              JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
                                                                ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
023222  000207                  RTS     PC                      ;  CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - WDPDR -
                        ;++ ************************************************************************
                        ;*                      - WRITE DATA PATTERN TO DEVICE REGISTERS -
                        ;*      THIS ROUTINE WRITES A ROTATED DATA PATTERN TO EACH OF THE 6 DEVICE
                        ;*      REGISTERS OF EACH ACTIVE LINE OF THE DEVICE UNDER TEST.
                        ;*      THE DATA PATTERN IS ROTATED ONCE AFTER EACH WRITE TO A DEVICE REGISTER
                        ;*              ON A PARTICULAR LINE.  THE STARTING DATA PATTERN FOR EACH LINE
                        ;*              IS ROTATED ONCE AFTER WRITING ALL THE REGISTERS ON A PARTICULAR
                        ;*              LINE.  THIS LEADS TO THE FOLLOWING DATA PATTERN:
                        ;*                      LINE 0, REGISTER 0 - SHIFTED 0 BIT POSITIONS
                        ;*                      LINE 0, REGISTER 1 - SHIFTED 1 BIT POSITION
                        ;*                      ...
                        ;*                      LINE 1, REGISTER 0 - SHIFTED 1 BIT POSITION
                        ;*                      LINE 2, REGISTER 1 - SHIFTED 2 BIT POSITIONS
                        ;*                      ...
                        ;*      ANY BITS FIELDS IN THE DEVICE REGISTERS THAT CANNOT BE ALTERED
                        ;*      ARE MASKED OUT OF THE DATA PATTERN BEFORE IT IS WRITTEN.
                        ;*      THIS ROUTINE WILL USE EITHER MOV, MOVB, BIS, BISB, BIC, OR BICB
                        ;*      INSTRUCTIONS.  THE UPPER OR LOWER BYTE CAN BE SPECIFIED FOR WRITING.
                        ;*
                        ;* INPUTS:      R2 - USED TO PASS IN THE DATA PATTERN TO BE ROTATED & WRITTEN.
                        ;*              R3 - BYTE INDICATOR (- => LO BYTE, + => HI BYTE, 0 => BOTH).
                        ;*              R4 - OPERATION TYPE INDICATOR (- => BIC, + => BIS, 0 => MOV).
                        ;*              ACTLNS - BIT MAP OF THE ACTIVE LINES ON THE DEVICE UNDER TEST.
                        ;*              CSRA - CONTAINS THE CSR ADDRESS OF THE DEVICE UNDER TEST.
                        ;*              DRADRT - BASE ADDRESS OF DEVICE REGISTER ADDRESS TABLE.
                        ;*              LPRO - EQUATED TO LPR REG OFFSET FROM DEVICE CSR ADDRESS.
                        ;*              NUMLNS - NUMBER OF LINES ON THE DEVICE UNDER TEST.
                        ;*              TXBFCO - EQUATED TO TBUFFCT REG OFFSET FROM DEVICE CSR ADDRESS.
                        ;*              UNBTTB - BASE ADDRESS OF THE UNUSED BIT TABLE.
                        ;*
                        ;* OUTPUTS:     DEVICE REGISTERS ON ALL ACTIVE DEVICE LINES ARE MODIFIED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WDPDR
                        ;*
                        ;* COMMENTS:    THIS ROUTINE DOES NOT WRITE DATA TO THE FOLLOWING REGISTERS.
                        ;*                              RBUF
                        ;*                              RXTIMER
                        ;*                              STAT
                        ;*                              FIFOSIZE
                        ;*                              FIFODATA
                        ;*              THE CSR IS CLEARED EXCEPT FOR THE IND.ADR.REG FIELD.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ROLDAP.
                        ;-- ************************************************************************
023224                  WDPDR:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
023224  004567  160542                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; SET UP OUTER LOOP WHICH WRITES THE DATA PATTERN TO EACH LINE'S REGISTERS
                        ;-
023230  005005                  CLR     R5              ;CLEAR LINE COUNTER TO SELECT LINE 0.
                        ;+
                        ; THE OUTER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP WRITES DATA TO ALL OF
                        ; THE DEVICE REGISTERS FOR A PARTICULAR LINE IF IT IS ACTIVE.
                        ;-
023232  010204          2$:     MOV     R2,R4           ;SAVE THE OUTER LOOP DATA PATTERN.
```

```
4112 023234  010577  156776                    MOV     R5,@CSRA          ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
4113 023240  006305                            ASL     R5                ;TURN LINE NUMBER INTO A WORD OFFSET.
4114 023242  036567  002404  156754            BIT     BITTBL(R5),ACTLNS
4115 023250  001456                           BEQ     20$               ;LINE ACTIVE?  NO, SKIP THIS LINE.
4116 023252  012701  000004                    MOV     #LPRO,R1          ;YES, INITIALIZE THE REGISTER OFFSET.
4117                                    ;+
4118                                    ;       THE INNER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP WRITES DATA TO A
4119                                    ;       DEVICE REGISTER.
4120                                    ;-
4121 023256  010200                    4$:     MOV     R2,R0
4122 023260  046100  002364                    BIC     UNBTTB(R1),R0     ;CLEAR BIT FIELDS FOR UNUSED REGISTER BITS.
4123 023264  016103  002236                    MOV     DRADRT(R1),R3     ;GET THE ADDRESS OF THE DEVICE REGISTER.
4124 023270  005766  000010                    TST     R3SLOT(SP)        ;CHECK THE OPERAND TYPE INDICATOR.
4125 023274  003402                           BLE     6$                ;HIGH BYTE?  NO, SKIP HIGH BYTE ADDRESS SET UP.
4126 023276  005203                            INC     R3                ;YES, SET THE REG ADDRESS TO THE HIGH BYTE.
4127 023300  000300                            SWAB    R0                ;MOVE HIGH BYTE DATA INTO THE LOW BYTE.
4128 023302  005766  000010            6$:     TST     R3SLOT(SP)        ;CHECK THE OPERAND TYPE INDICATOR.
4129 023306  001412                           BEQ     12$               ;WORD ACCESS?  YES, GO PERFORM WORD ACCESS.
4130                                        ;+
4131                                        ;PERFORM BYTE ACCESS TO THE SPECIFIED BYTE OF THE SPECIFIED REGISTER.
4132                                        ;-
4133 023310  005766  000012                    TST     R4SLOT(SP)        ;NO, CHECK THE ACCESS TYPE INDICATOR.
4134 023314  100403                           BMI     8$                ;USE BIC?  YES, GO PERFORM BICB INSTRUCTION.
4135 023316  001404                           BEQ     10$               ;USE MOV?  YES, GO PERFORM MOVB INSTRUCTION.
4136 023320  150013                            BISB    R0,(R3)           ;NEITHER.  PERFORM BISB ACCESS TO REGISTER.
4137 023322  000415                           BR      18$
4138 023324  140013                    8$:     BICB    R0,(R3)           ;PERFORM BICB ACCESS TO REGISTER.
4139 023326  000413                           BR      18$
4140 023330  110013                    10$:    MOVB    R0,(R3)           ;PERFORM MOVB ACCESS TO REGISTER.
4141 023332  000411                           BR      18$
4142                                        ;+
4143                                        ;PERFORM WORD ACCESS TO THE SPECIFIED REGISTER.
4144                                        ;-
4145 023334  005766  000012            12$:    TST     R4SLOT(SP)        ;CHECK THE ACCESS TYPE INDICATOR.
4146 023340  100403                           BMI     14$               ;USE BIC?  YES, GO PERFORM BIC INSTRUCTION.
4147 023342  001404                           BEQ     16$               ;USE MOV?  YES, GO PERFORM MOV INSTRUCTION.
4148 023344  050013                            BIS     R0,(R3)           ;NEITHER.  PERFORM BIS ACCESS TO REGISTER.
4149 023346  000403                           BR      18$
4150 023350  040013                    14$:    BIC     R0,(R3)           ;PERFORM BIC ACCESS TO REGISTER.
4151 023352  000401                           BR      18$
4152 023354  010013                    16$:    MOV     R0,(R3)           ;PERFORM MOV ACCESS TO REGISTER.
4153                                        ;+
4154                                        ; PREPARE THE DATA PATTERN AND OFFSET FOR THE NEXT REGISTER ON THIS LINE.
4155                                        ;-
4156 023356  004767  176042            18$:    JSR     PC,ROLDAP         ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
4157 023362  062701  000002                     ADD     #2,R1             ;INCREMENT OFFSET FOR NEXT REGISTER.
4158 023366  020127  000006                     CMP     R1,#FSLSO         ;CHECK IF THIS IS THE FIFOSIZE/DATA REG
4159 023372  001002                           BNE     19$               ;AVOID ALTERING THE OFFSET IF IT ISN'T.
4160 023374  062701  000002                     ADD     #2,R1             ;AVOID TESTING THESE REGISTERS.
4161 023400  020127  000016            19$:    CMP     R1,#TXBFCO        ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
4162 023404  003724                           BLE     4$                ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
4163                                        ;+
4164                                        ; BACK INTO THE OUTER LOOP.  NOW SET UP FOR NEXT LINE.  LOOP IF NOT DONE.
4165                                        ;-
4166 023406  010402                    20$:    MOV     R4,R2             ;SET UP TO ROTATE THE DATA PATTERN.
4167 023410  004767  176010                    JSR     PC,ROLDAP         ;ROTATE THE DATA PATTERN.
4168 023414  006205                            ASR     R5                ;CONVERT BACK TO LINE NUMBER FROM WORD OFFSET.
```

```
 4169 023416  005205                    INC     R5              ;COUNT THIS LINE.
 4170 023420  020527  000020            CMP     R5,#NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
 4171 023424  002702                    BLT     2$              ;LOOP IF SOME LINES NOT DONE.
 4172
 4173 023426                    60$:    PASS                    ;RESTORE GPRS.
      023426  004736                            JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
 4174 023430  000207                    RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - WTWLNC -
                        ;++ ************************************************************************
                        ;*                        - LINE CONTROL REGISTER SETUP ROUTINE -
                        ;*      THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
                        ;*      CONTROL REGISTERS (LNCTRL) TO THE SPECIFIED STATE.  ONLY THE LNCTRLS
                        ;*      FOR THE SPECIFIED LINES ARE ALTERED.
                        ;*
                        ;* INPUTS:      R0 - NEW LINE PARAMETERS.
                        ;*              R5 - BIT MAP OF LINES TO BE ALTERED.
                        ;*              CSRA - CONTAINS ADDRESS OF THE DUT CSR.
                        ;*              IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
                        ;*               ENABLE BITS IN THE CSR.
                        ;*              LNCTRA - CONTAINS ADDRESS OF THE DUT LNCTRL REGISTERS.
                        ;*
                        ;* OUTPUTS:     LNCTRL - SPECIFIED DUT LINE CONTROL REGISTERS ARE ALTERED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WTWLNC
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
                        ;-- ************************************************************************

023432                  WTWLNC:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023432  004567  160334                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
                        ;-
023436  016701  156604          MOV     LNCTRA,R1               ;SET UP THE REGISTER ADDRESS PARAMETER.
023442  010002                  MOV     R0,R2                   ;SET UP THE DESIRED REGISTER CONTENTS.
023444  010503                  MOV     R5,R3                   ;SET UP THE BIT MAP OF LINES TO ALTER.
023446  012704  177777          MOV     #-1,R4                  ;SELECT ALL REGISTER BITS TO BE ALTERED.
                        ;+
                        ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
                        ;-
023452  004767  173732          JSR     PC,ALTFLD               ;ALTER THE REGISTER CONTENTS.

023456                  60$:    PASS                            ;RESTORE GPRS.
023456  004736                          JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
023460  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - WTWLPR -
                                ;++ ************************************************************************
                                ;*                        - LINE PARAMETER REGISTER SETUP ROUTINE -
                                ;*      THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
                                ;*      PARAMETER REGISTERS (LPR) TO THE SPECIFIED STATE.  ONLY THE LPRS FOR
                                ;*      THE SPECIFIED LINES ARE ALTERED.
                                ;*
                                ;* INPUTS:      R0 - NEW LINE PARAMETERS.
                                ;*              R5 - BIT MAP OF LINES TO BE ALTERED.
                                ;*              CSRA - CONTAINS ADDRESS OF THE DUT CSR.
                                ;*              IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
                                ;*               ENABLE BITS IN THE CSR.
                                ;*              LPRA - CONTAINS ADDRESS OF THE DUT LPR.
                                ;*
                                ;* OUTPUTS:     LPR - SPECIFIED DUT LINE PARAMTER REGISTERS ARE ALTERED.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,WTWLPR
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
                                ;-- ************************************************************************

023462                          WTWLPR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023462  004567  160304                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;+
                                ; SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
                                ;-
023466  016701  156550                  MOV     LPRA,R1                 ;SET UP THE REGISTER ADDRESS PARAMETER.
023472  010002                          MOV     R0,R2                   ;SET UP THE DESIRED REGISTER CONTENTS.
023474  010503                          MOV     R5,R3                   ;SET UP THE BIT MAP OF LINES TO ALTER.
023476  012704  177777                  MOV     #-1,R4                  ;SELECT ALL REGISTER BITS TO BE ALTERED.
                                ;+
                                ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
                                ;-
023502  004767  173702                  JSR     PC,ALTFLD               ;ALTER THE REGISTER CONTENTS.

023506                          60$:    PASS                            ;RESTORE GPRS.
023506  004736                                  JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
023510  000207                          RTS     PC
```

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

023512                  CACHRX::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
023512  004567  160254                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023516  016701  156566          MOV     RXINTC,R1       ;GET THE RECEIVER INTERRUPT COUNT
023522  005201                  INC     R1              ;INCREMENT THE COUNT
023524  102001                  BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED
023526  005301                  DEC     R1              ;RESET THE COUNT TO 177777
023530  010167  156554  2$:     MOV     R1,RXINTC       ;SAVE NEW COUNT VALUE
023534                  60$:    PASS                    ;RESTORE GPRS.
023534  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023536  000002                  RTI
```

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

023540                  CACHTX::SAVE                    ;SAVE CONTENTS OF GPRS RO THRU R5.
023540  004567  160226                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023544  016701  156552          MOV     TXINTC,R1       ;GET THE TRANSMISSION INTERRUPT COUNT
023550  005201                  INC     R1              ;INCREMENT THE COUNT
023552  102001                  BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED
023554  005301                  DEC     R1              ;RESET THE COUNT TO 177777
023556  010167  156540  2$:     MOV     R1,TXINTC       ;SAVE NEW COUNT VALUE
023562                  60$:    PASS                    ;RESTORE GPRS.
023562  004736                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023564  000002                  RTI
```

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

023566  005767  156546        CLKINT:: TST     TIMER1          ;CHECK FOR TIMER1 AT ZERO.
023572  001402                         BEQ     2$              ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
023574  005367  156540                 DEC     TIMER1          ;DECREMENT TIME COUNT.
023600  005767  156536        2$:      TST     TIMER2          ;CHECK FOR TIMER2 AT ZERO.
023604  001402                         BEQ     4$              ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
023606  005367  156530                 DEC     TIMER2          ;DECREMENT TIME COUNT.
023612  005367  156526        4$:      DEC     TIMER3          ;DECREMENT THE BREAK COUNT.
023616  001006                         BNE     60$             ;EXIT IF NOT TIME TO CALL BREAK.
023620  016767  156522  156516        MOV     BCOUNT,TIMER3   ;SET UP TIME TILL NEXT BREAK.
023626  010046                         MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 FROM BREAK MACRO.
023630                                 BREAK                   ;CHECK FOR OPERATOR CONTROL/C.
023630  104422                                                         TRAP    C$BRK
023632  012600                         MOV     (SP)+,R0        ;RESTORE CONTENTS OF R0.
023634  000002          60$:           RTI
```

```
                        .SBTTL  INTERUPT SERVICE ROUTINE        - RXBRRT -
                        ;++ ************************************************************************
                        ;*      - BR LEVEL TEST RECEIVE INTERRUPT SERVICE ROUTINE -
                        ;*      THIS SERVICE ROUTINE HANDLES RECEIVE INTERRUPTS DURING THE INTERRUPT
                        ;*      BR LEVEL TEST.  THIS ROUTINE COUNTS THE INTERRUPT AND SETS A FLAG
                        ;*      TO INDICATE THAT THE INTERRUPT HAS OCCURRED.  IT ALSO CHECKS THE
                        ;*      FLAG WHICH INDICATES THAT A TX INTERRUPT HAS OCCURRED.  IF THE TX
                        ;*      INTERRUPT FLAG IS SET, THIS ROUTINE SETS AN INTERRUPT ORDER ERROR
                        ;*      FLAG INDICATING THAT A TRANSMIT INTERRUPT WAS SERVICED BEFORE A
                        ;*      SIMULTANEOUS RECEIVE INTERRUPT.
                        ;*
                        ;* INPUTS:      RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERUPTS.
                        ;*              RXINTF - RX INTERRUPT FLAGS.
                        ;*
                        ;* OUTPUTS:     RXINTC - CONTAINS THE UPDATED INTERUPT COUNT.
                        ;*              RXINTF - RX INT FLAGS:
                        ;*                       (BIT 0 SET, BIT 14 SET IF TXINTF BIT 0 IS SET.)
                        ;*
                        ;* CALLING SEQUENCE:    PUT THE ADDRESS OF THE LABEL RXBRRT IN THE VECTOR
                        ;*                      LOCATION.
                        ;*
                        ;* COMMENTS:    NOTE: THE FIFO IS NOT PURGED BY THIS ROUTINE.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

023636                  RXBRRT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
023636  004567  160130                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023642  017700  156372          MOV     @RBUFA,R0       ;READ THE CHAR OUT OF THE FIFO.
023646  016701  156436          MOV     RXINTC,R1       ;GET THE INTERUPT COUNT.
023652  005201                  INC     R1              ;INCREMENT THE COUNT.
023654  001402                  BEQ     2$              ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
023656  010167  156426          MOV     R1,RXINTC       ;SAVE NEW COUNT VALUE.
023662  016701  156424  2$:     MOV     RXINTF,R1       ;GET THE RX INTERRUPT FLAGS.
023666  052701  000001          BIS     #BIT0,R1        ;SET THE RX INTERRUPT HAS OCCURRED FLAG.
023672  032767  000001  156424  BIT     #BIT0,TXINTF    ;TEST THE "TX INT HAS OCCURRED" FLAG.
023700  001402                  BEQ     4$              ;SKIP SETTING ERROR FLAG IF NO TX INT.
023702  052701  040000          BIS     #BIT14,R1       ;SET THE INTERRUPT ORDER ERROR FLAG.
023706  010167  156400  4$:     MOV     R1,RXINTF       ;UPDATE THE RX INTERRUPT FLAGS.
023712                  60$:    PASS                    ;RESTORE GPRS.
023712  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023714  000002                  RTI
```

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

023716                  RXINPT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023716  004567  160050                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023722  017701  156310          MOV     @CSRA,R1                ;READ THE CONTENTS OF THE CSR.
023726  032701  000200          BIT     #BIT7,R1                ;TEST RX.DATA.AVAIL BIT.
023732  001003                  BNE     2$                      ;BRANCH AROUND SETTING FLAG IF BIT IS SET.
023734  052767  100000  156350  BIS     #BIT15,RXINTF           ;SET THE RX.DATA.AVAIL CLEAR FLAG.
023742  016701  156342  2$:     MOV     RXINTC,R1               ;GET THE INTERUPT COUNT.
023746  005201                  INC     R1                      ;INCREMENT THE COUNT.
023750  001402                  BEQ     4$                      ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
023752  010167  156332          MOV     R1,RXINTC               ;SAVE NEW COUNT VALUE.
023756  016702  156314  4$:     MOV     BUFPTR,R2               ;GET THE POINTER TO NEXT FREE BUFFER WORD.
023762  017722  156252          MOV     @RBUFA,(R2)+            ;READ A CHAR FROM THE FIFO INTO BUFFER.
023766  020267  157730          CMP     R2,BUFEND               ;TEST FOR POINTER BEYOND END OF BUFFER.
023772  103002                  BHIS    60$                     ;SKIP THE PTR UPDATE IF PTR OUT OF BOUNDS.
023774  010267  156276          MOV     R2,BUFPTR               ;UPDATE THE BUFFER POINTER.
024000                  60$:    PASS                            ;RESTORE GPRS.
024000  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
024002  000002                  RTI
```

```
                                .SBTTL  GLOBAL TRAP SERVICE ROUTINE     - TP4RTN -
                                ;*******************************************************************************
                                ;*      BUS TIME-OUT TRAP (004 TRAP) SERVICE ROUTINE -
                                ;*      THIS ROUTINE 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.
                                ;*******************************************************************************

024004  021627  017720        TP4RTN:: CMP     (SP),@ADRPTR    ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
024010  001402                        BEQ     2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
024012  000177  156300                JMP     @TP4VEC         ;IF NOT,JUMP TO NORMAL 004 TRAP SERVICE RTN.
024016  052767  100000  156270  2$:   BIS     #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
024024  000002                        RTI                     ;ALL DONE, GO BACK TO THE TEST.
```

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

024026                          TXINTR:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
024026  004567  157740                         JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
024032  016701  156264                 MOV     TXINTC,R1       ;GET THE TX INTERUPT COUNT.
024036  005201                         INC     R1              ;INCREMENT THE COUNT.
024040  102001                        BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED.
024042  005301                         DEC     R1              ;RESET THE COUNT TO 177777.
024044  010167  156252          2$:    MOV     R1,TXINTC       ;SAVE NEW COUNT VALUE.
024050  016703  156250                 MOV     TXINTF,R3       ;GET THE TX INTERRUPT FLAGS.
024054  017702  156156                 MOV     @CSRA,R2        ;READ THE CSR.
024060  100402                        BMI     4$              ;SKIP SETTING OF FLAG IF TX.ACTION IS SET.
024062  052703  100000                 BIS     #BIT15,R3       ;SET THE TX.ACTION CLEAR FLAG.
024066  052703  000001          4$:    BIS     #BIT0,R3        ;SET THE TX INT HAS OCCURRED FLAG.
024072  010367  156226                 MOV     R3,TXINTF       ;UPDATE THE TX INTERRUPT FLAGS.
024076                          60$:    PASS                    ;RESTORE GPRS.
024076  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
024100  000002                        RTI
```

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


                        .SBTTL  REPORT CODING SECTION

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

024102                          BGNRPT
024102                                                          L$RPT::

                        EXIT    RPT
024102  000167                                                         .WORD   J$JMP
024104  000000                                                         .WORD   L10017-2-.

                        .EVEN

                        ENDRPT
024106                                                          L10017:
024106  104425                                                         TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE
                                ;*****************************************************************************
*
                                ;
                                ;               FVTSKL4.P11
                                ;
                                ;*****************************************************************************


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

024110                                  BGNPROT
024110                                                                  L$PROT::

024110  177777                         -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
024112  177777                         -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
024114  177777                         -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

024116                                  ENDPROT
```

```
                        ;*****************************************************************************
                        ;
                        ;               FVTA.INI
                        ;
                        ;*****************************************************************************



                        .SBTTL  INITIALIZE SECTION
                        ;++
                        ;*****************************************************************************
                        ;*      THIS SECTION CONTAINS THE CODE WHICH IS PERFORMED AT THE BEGINNING OF
                        ;*      EACH PASS OR AFTER A CONTINUE COMMAND.
                        ;*      THIS CODE PERFORMS THE FOLLOWING ACTIONS:
                        ;*
                        ;*      MOVES THE INFORMATION HELD IN THE HARDWARE P-TABLE INTO THE GLOBAL
                        ;*      DATA AREA.
                        ;*
                        ;*****************************************************************************
                        ;--
                                BGNINIT
                                                                                L$INIT::
                        ;SEE IF PROGRAM JUST STARTED, BR IF YES
                                READEF  #EF.START
024116  012700  000040                                                          MOV     #EF.START,R0
024122  104447                                                                  TRAP    C$REFG
                                BCOMPLETE       NEWSTA
024124  103416                                                                  BCS     NEWSTA
                        ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
                                READEF  #EF.RESTART
024126  012700  000037                                                          MOV     #EF.RESTART,R0
024132  104447                                                                  TRAP    C$REFG
                                BCOMPLETE       NEWRES
024134  103556                                                                  BCS     NEWRES
                        ;SEE IF THIS IS A NEW PASS, BR IF YES
                                READEF  #EF.NEW
024136  012700  000035                                                          MOV     #EF.NEW,R0
024142  104447                                                                  TRAP    C$REFG
                                BCOMPLETE       NEWPAS
024144  103555                                                                  BCS     NEWPAS
                        ;SEE IF PROGRAM WAS JUST CONTINUED
                                READEF  #EF.CONTINUE
024146  012700  000036                                                          MOV     #EF.CONTINUE,R0
024152  104447                                                                  TRAP    C$REFG
                                BNCOMPLETE      GETPRM
024154  103161                                                                  BCC     GETPRM
                                JMP     ENDIT
                        NEWSTA:
                                BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
024162  104433                                                                  TRAP    C$RESET
                        ;+
                        ; SET UP FOR LINE TIME CLOCK INTERRUPTS.
                        ;-
                                CLOCK   L,R1            ;GET THE CLOCK PARAMETERS.
024164  012700  000114                                                          MOV     #'L,R0
024170  104462                                                                  TRAP    C$CLCK
```

INITIALIZE SECTION

```
                024172  010001                                                          MOV     R0,R1
4610            024174  012167  156130                  MOV     (R1)+,CLKCSR            ;STORE CLOCK CSR ADDRESS.
4611            024200  012167  156126                  MOV     (R1)+,CLKBRL            ;STORE CLOCK BUS REQ INT LEVEL.
4612            024204  012167  156124                  MOV     (R1)+,CLKVEC            ;STORE CLOCK INTERRUPT VECTOR.
4613            024210  012167  156122                  MOV     (R1)+,CLKHRZ            ;STORE CLOCK FREQUENCY.
4614            024214  026727  156116  000062          CMP     CLKHRZ,#50.             ;TEST FOR 50HZ LINE FREQUENCY.
4615            024222  001004                          BNE     2$                      ;BRANCH IF CLOCK IS NOT 50HZ.
4616            024224  012767  000024  156116          MOV     #20.,MSTICK             ;INDICATE 20MS PER CLOCK TICK.
4617            024232  000403                          BR      4$
4618            024234  012767  000021  156106  2$:     MOV     #17.,MSTICK             ;INDICATE 17 MS PER CLOCK TICK.
4619            024242                          4$:     SETVEC  CLKVEC,#CLKINT,PRI06    ;INITIALIZE CLOCK INTERRUPT VECTOR.
                024242  016746  154032                                                  MOV     PRI06,-(SP)
                024246  012746  023566                                                  MOV     #CLKINT,-(SP)
                024252  016746  156056                                                  MOV     CLKVEC,-(SP)
                024256  012746  000003                                                  MOV     #3,-(SP)
                024262  104437                                                          TRAP    C$SVEC
                024264  062706  000010                                                  ADD     #10,SP
4620            024270  016700  156042                  MOV     CLKHRZ,R0               ;INITIALIZE THE BREAK COUNT
4621            024274  006300                          ASL     R0                      ; TO CAUSE A BREAK
4622            024276  010067  156044                  MOV     R0,BCOUNT               ; EVERY 2 SECONDS.
4623            024302                                  SETPRI  #PRI05                  ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
                024302  012700  000240                                                  MOV     #PRI05,R0
                024306  104441                                                          TRAP    C$SPRI
4624                                            ;+
4625                                            ; ENABLE THE LINE TIME CLOCK (LTC) CHECKING TO MAKE SURE THAT THE CSR
4626                                            ; IS ACCESSABLE.
4627                                            ;    FIRST SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
4628                                            ;-
4629            024310  016767  153470  156000          MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
4630            024316  012767  024004  153460          MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4631
4632                                            ;+
4633                                            ;    ENABLE LTC CHECKING FOR 004 TRAP IN CASE CSR IS NOT THERE.
4634                                            ;-
4635            024324  005067  155764          CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
4636            024330  012767  000100  155770  MOV     #BIT6,WORD1     ;SET UP TO SET BIT6 OF THE LTC CSR.
4637            024336  012700  002326          MOV     #WORD1,R0       ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
4638            024342  016701  155762          MOV     CLKCSR,R1       ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
4639            024346  004767  173334          JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
4640            024352  016767  155740  153424  MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4641            024360  103403                  BCS     6$              ;IF NO TRAP, LTC IS THERE SO CONTINUE.
4642            024362  005067  155750          CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
4643            024366  000402                  BR      8$              ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
4644                                            ;+
4645                                            ; CALIBRATE THE DELAY ROUTINE MILLI-SECOND DELAY COUNT VALUE.
4646                                            ;-
4647            024370  004767  173066  6$:     JSR     PC,CALMSL
4648                                            ;+
4649                                            ; CHECK FOR MEMMORY MANAGEMENT PRESENT ON THIS MACHINE.
4650                                            ; IF MEM MGT IS PRESENT, DISABLE IT.
4651                                            ;-
4652            024374  016767  153404  155714  8$:     MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
4653            024402  012767  024004  153374          MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4654            024410  005067  155700                  CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
4655            024414  005067  155706                  CLR     WORD1           ;PREPARE TO CLEAR THE MEM MGT SR0 REGISTER.
4656            024420  012700  002326                  MOV     #WORD1,R0       ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
4657            024424  016701  155724                  MOV     MMSR0,R1        ;SELECT MEM MGT SR0 REGISTER AS DESTINATION.
4658            024430  005067  155722                  CLR     MMPRES          ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
```

```
4658 024434  005067  155720                    CLR     MMENAB          ;INDICATE MEM MGT IS NOT ENABLED.
4659 024440  004767  173242                    JSR     PC,CKTRAP       ;CLEAR THE MEM MCT SR0 REG AND CHECK FOR TRAP.
4660 024444  016767  155646  153332            MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4661 024452  103003                            BCC     10$             ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
4662 024454  012767  000001  155674            MOV     #1,MMPRES       ;INDICATE THAT MEM MGT IS PRESENT.
4663 024462  005067  155620          10$:      CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4664 024466  000167  000006                    JMP     NEWPAS          ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
4665
4666 024472                          NEWRES:   BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
     024472  104433                                                                  TRAP    C$RESET
4667 024474  005067  155606                    CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4668 024500                          NEWPAS:
4669 024500  012767  177777  155524            MOV     #-1,UNITN       ;RESET LOGICAL DEVICE TO -1
4670                                            ;+
4671                                            ; INCREMENT THE PASS COUNTER, CORRECT FOR ANY OVERFLOW.
4672                                            ;  THIS COUNTER IS USED IN THE ROM VERSION TEST.
4673                                            ;-
4674 024506  005267  155574                    INC     PASCNT          ;INCREMENT THE PASS COUNTER.
4675 024512  001002                            BNE     GETPRM          ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
4676 024514  005367  155566                    DEC     PASCNT          ;SET PASS COUNT TO 177777 OCTAL.
4677
4678                                            ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
4679 024520                          GETPRM:
4680 024520  005267  155506                    INC     UNITN           ;INCREMENT LOGICAL DEVICE NUMBER
4681 024524  026767  155502  155260            CMP     UNITN,L$UNIT    ;SEE IF MAXIMUM UNIT NO. EXCEEDED
4682 024532  002362                            BGE     NEWPAS          ;BR IF YES
4683
4684 024534                                    GPHARD  UNITN,R1        ;GET P-TABLE POINTER INTO R1
     024534  016700  155472                                                          MOV     UNITN,R0
     024540  104442                                                                  TRAP    C$GPHRD
     024542  010001                                                                  MOV     R0,R1
4685 024544                                    BCOMPLETE       30$     ;BR IF DEVICE AVAILABLE
     024544  103401                                                                  BCS     30$
4686 024546  000764                            BR      GETPRM          ;SKIP THIS DEVICE
4687
4688
4689                                            ;************ HARDWARE PARAMETER MOVING CODE *****************
4690 024550  012167  155462          30$:      MOV     (R1)+,CSRA      ;STORE DHU-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
4691 024554  012102                            MOV     (R1)+,R2        ;GET THE RX INTERRUPT VECTOR ADDRESS.
4692 024556  010267  155444                    MOV     R2,RXVECA       ;STORE RX INT VECTOR ADDRESS.
4693 024562  062702  000004                    ADD     #4,R2           ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
4694 024566  010267  155436                    MOV     R2,TXVECA       ;STORE TX INT VECTOR ADDRESS.
4695 024572  012167  155426                    MOV     (R1)+,ACTLNS    ;STORE DHU-11 ACTIVE LINE BIT MAP
4696 024576  111167  155432                    MOVB    (R1),BRLEVL     ;STORE DHU-11 INTERUPT BUS REQUEST LEVEL
4697
4698                                            ;+
4699                                            ;       CALCULTATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
4700                                            ;       DEVICE REGISTER ADDRESS TABLE.
4701 024602  016701  155430                    ;-      MOV     CSRA,R1         ;COPY CSR ADDRESS
4702 024606  005201                            MOV     R1              ;INCREMENT CSR ADDRESS
4703 024610  005201                            INC     R1              ; COPY BY 2.
4704 024612  012703  000007                    INC     R1              ;SET UP REGISTER COUNT
4705 024616  012702  002240                    MOV     #7,R3           ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
4706 024622  010122                    12$:    MOV     #RBUFA,R2       ;STORE REGISTER ADDRESS IN TABLE
4707 024624  005201                            MOV     R1,(R2)+        ;INCREMENT REGISTER ADDRESS
4708 024626  005201                            INC     R1              ; BY 2,FOR THE NEXT DEVICE REGISTER.
4709 024630  005303                            INC     R1              ;DECREMENT REGISTER COUNT
                                               DEC     R3
```

```
4710 024632  001373                        BNE     12$             ;LOOP IF NOT DONE
4711
4712                                ;+
4713                                ; INITIALISE THE BMP CODE QUEUE.
4714                                ;-
4715 024634  012700  002522                 MOV     #BMPCQB,R0      ;GET THE START ADDRESS OF THE QUEUE.
4716 024640  012701  002722                 MOV     #BMPCQE,R1      ;GET THE END ADDRESS OF THE QUEUE.
4717 024644  010067  155650                 MOV     R0,BMPCQP       ;SET THE POINTER TO THE START OF THE QUEUE.
4718 024650  005020                 14$:    CLR     (R0)+           ;CLEAR OUT THE CONTENTS OF THE QUEUE.
4719 024652  020001                         CMP     R0,R1           ;CHECK IF END OF QUEUE HAS BEEN REACHED.
4720 024654  103775                        BLO     14$             ;LOOP IF NOT ALL DONE.
4721                                ;+
4722                                ; REPORT THE UNIT NUMBER IF THE SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
4723                                ;  AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
4724                                ;-
4725 024656  032767  000020  155334          BIT     #BIT4,OPTION    ;CHECK IF THE QUESTION WAS ANSWERED YES.
4726 024664  001416                        BEQ     16$             ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
4727 024666  026727  155120  000001         CMP     L$UNIT,#1       ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
4728 024674  003412                        BLE     16$             ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
4729 024676                                PRINTF  #MFUNIT,UNITN   ;REPORT UNIT NUMBER.
     024676  016746  155330                                                    MOV     UNITN,-(SP)
     024702  012746  004070                                                    MOV     #MFUNIT,-(SP)
     024706  012746  000002                                                    MOV     #2,-(SP)
     024712  010600                                                            MOV     SP,R0
     024714  104417                                                            TRAP    C$PNTF
     024716  062706  000006                                                    ADD     #6,SP
4730 024722                        16$:
4731
4732 024722  005067  155354        ENDIT:  CLR     CTRLCF          ;CLR THE CTRL-C TEST ABORT FLAG.
4733                                ;+
4734                                ; SET THE PROCESSOR PRIORITY TO ALLOW LTC INTERRUPTS BUT NOT OTHERS.
4735                                ;-
4736 024726                                SETPRI  #PRI07                  ;SET PROCESSOR PRIORITY TO 7.
     024726  012700  000340                                                    MOV     #PRI07,R0
     024732  104441                                                            TRAP    C$SPRI
4737 024734                                ENDINIT
     024734                                                             L10021:
     024734  104411                                                            TRAP    C$INIT
4738
4739            000000                      TNUM == 0               ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.
```

```
                        ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        ;
                        ;               FVTA.ATD
                        ;
                        ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


                        .SBTTL  AUTODROP SECTION


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

                                BGNAUTO
024736                                                          L$AUTO::

                        ENDAUTO
024736                                                          L10022:
024736  104461                                                          TRAP    C$AUTO
```

```
4771
4772                                    ;*****************************************************************************
4773                                    ;
4774                                    ;               FVT.CUC
4775                                    ;
4776                                    ;*****************************************************************************
4777
4778
4779
4780                                    .SBTTL  CLEANUP CODING SECTION
4781
4782                                    ;++
4783                                    ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4784                                    ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
4785                                    ;--
4786
4787 024740                                     BGNCLN
     024740                                                                    L$CLEAN::
4788
4797 024740  005767  155336                      TST     CTRLCF          ;DID WE GET HERE BY CTRL-C FROM TEST?
4798 024744  001401                              BEQ     2$              ;CTRL-C FROM TEST?  NO, SKIP BUS RESET.
4799 024746                                      BRESET                  ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
     024746  104433                                                              TRAP    C$RESET
4800 024750                             2$:
4801 024750                                     EXIT    CLN
     024750  104432                                                              TRAP    C$EXIT
     024752  000002                                                              .WORD   L10023-.
4802
4814
4815                                            .EVEN
4816
4817 024754                                     ENDCLN
     024754                                                                  L10023:
     024754  104412                                                              TRAP    C$CLEAN
```

```
4819
4820                            ;*****************************************************************************************
4821                            ;
4822                            ;               FVTA.DRP
4823                            ;
4824                            ;*****************************************************************************************
4825
4826
4827
4828                            .SBTTL  DROP UNIT SECTION
4829
4830                            ;++
4831                            ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4832                            ; TO NO LONGER BE TESTED.
4833                            ;--
4834
4835 024756                             BGNDU
      024756                                                                  L$DU::
4836
4837                            ;*****************************************************************************************
4838                            ;       INSERT DROP CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
4839                            ;       A "DROP" COMMAND OR A "DODU" MACRO EXECUTION.  THE PURPOSE
4840                            ;       OF THIS CODE IS TO DO ANY NECESSARY HOUSEKEEPING AFTER A
4841                            ;       UNIT HAS BEEN DROPPED.  THIS SECTION IS OPTIONAL.
4842                            ;*****************************************************************************************
4843
4844 024756                             PRINTF #DROP,R0        ;REPORT UNIT THAT HAS BEEN DROPPED.
      024756  010046                                                          MOV     R0,-(SP)
      024760  012746  025002                                                  MOV     #DROP,-(SP)
      024764  012746  000002                                                  MOV     #2,-(SP)
      024770  010600                                                          MOV     SP,R0
      024772  104417                                                          TRAP    C$PNTF
      024774  062706  000006                                                  ADD     #6,SP
4845 025000  000427                     BR      EDROP          ;BRANCH AROUND THE MESSAGE.
4846
4847 025002      045     101     040  DROP:   .ASCIZ/%A UNIT%D6%A DROPPED FROM FURTHER TESTING.%N/
      025005      125     116     111
      025010      124     045     104
      025013      066     045     101
      025016      040     104     122
      025021      117     120     120
      025024      105     104     040
      025027      106     122     117
      025032      115     040     106
      025035      125     122     124
      025040      110     105     122
      025043      040     124     105
      025046      123     124     111
      025051      116     107     056
      025054      045     116     000
4848                            .EVEN
4849 025060                     EDROP:
4850 025060                             EXIT    DU
      025060  000167                                                          .WORD   J$JMP
      025062  000000                                                          .WORD   L10024-2-.
4851
4852
```

DROP UNIT SECTION

```
4853
4854 025064                            ENDDU
     025064                                                  L10024:
     025064  104453                                                TRAP    C$DU
```

```
                        ;*****************************************************************************
                        ;
                        ;               FVTA.ADD
                        ;
                        ;*****************************************************************************



                        .SBTTL  ADD UNIT SECTION

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

025066                          BGNAU
025066                                                                  L$AU::

                        ;*****************************************************************************
                        ;       INSERT ADD CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
                        ;       AN "ADD" COMMAND.  THE PURPOSE OF THIS CODE IS TO DO ANY
                        ;       HOUSEKEEPING THAT MAY BE NECESSARY AFTER A UNIT HAS BEEN ADDED.
                        ;       THIS SECTION IS OPTIONAL.
                        ;*****************************************************************************

025066                          EXIT    AU
025066  000167                                                         .WORD   J$JMP
025070  000000                                                         .WORD   L10025-2-.

                        .EVEN

025072                  ENDAU
025072                                                          L10025:
025072  104452                                                          TRAP    C$AU
```

```
                                .SBTTL  HARDWARE TEST           - ADRA -
                                ;++
                                ;*****************************************************************************
                                ;*                      - REGISTER ADDRESS TEST -
                                ;*
                                ;*      THIS TEST VERIFIES THAT THE DEVICE REGISTERS WILL RESPOND TO THE PROPER
                                ;*      UNIBUS HANDSHAKING SIGNALS WHEN ACCESSED. IF THE DHU11 DOES NOT RESPOND
                                ;*      TO THE ACCESS ATTEMPTS (IF THE DHU11 IS AT THE WRONG ADDRESS, FOR EXAMPLE)
                                ;*      THE 004 BUS TIME-OUT TRAP IS DETECTED BY THIS ROUTINE AND AN ERROR
                                ;*      IS REPORTED. THIS TEST IS PERFORMED ON LINE 0 ONLY.
                                ;*
                                ;*****************************************************************************
                                ;--

025074                                  BGNTST
025074                                                                  T1::
        000001                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025074  012767  000001  155216           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (1)
025102  012767  177777  155172           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
025110  012767  000145  156646           MOV    #101.,ERRNBR    ;SET THE TEST ERROR NUMBER IN THE TABLE.
025116  012767  005564  156642           MOV    #EM0103,ERRMSG  ;SET UP THE TEST FAILURE MESSAGE IN THE TABLE.
025124  012767  015444  156636           MOV    #ER0101,ERRBLK  ;SET-UP THE ERROR ROUTINE IN THE ERROR TABLE.
                                ;+
                                ; SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
                                ;-
025132  016767  152646  155156           MOV    4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
025140  012767  024004  152636           MOV    #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
025146  005005                           CLR    R5              ;CLEAR THE ERROR FLAGS.

                                ;+
                                ; 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.
                                ;-
025150  016700  155062                   MOV    CSRA,R0         ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
025154  012701  025346                   MOV    #52$,R1         ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
025160  004767  172522                   JSR    PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
025164  103402                           BCS    4$              ;IF NO TRAP, BYPASS ERROR.
025166  052705  100001                   BIS    #100001,R5      ;SET FATAL READ ERROR FLAGS.
025172  042767  000017  000146  4$:      BIC    #17,52$         ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
025200  010100                           MOV    R1,R0           ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
025202  016701  155030                   MOV    CSRA,R1         ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
025206  004767  172474                   JSR    PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
025212  103403                           BCS    6$              ;IF NO TRAP, BYPASS ERROR.
025214  052705  100002                   BIS    #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
025220  000434                           BR     40$             ;EXIT AND REPORT FATAL ERROR.
                                ;+
                                ;    NOW, WE TEST EACH REGISTER FOR THIS LINE.
                                ;-
025222  012702  000010           6$:     MOV    #8.,R2          ;INIT REGISTER COUNTER TO 8.
025226  016767  155004  000110           MOV    CSRA,50$        ;INITIALIZE THE REGISTER POINTER.
025234  012700  025344           8$:     MOV    #50$,R0         ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
025240  012701  025346                   MOV    #52$,R1         ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
025244  004767  172436                   JSR    PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
025250  103402                           BCS    10$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
025252  052705  100001                   BIS    #100001,R5      ;SET FATAL READ ERROR FLAGS.
025256  010100                  10$:     MOV    R1,R0           ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
025260  012701  025344                   MOV    #50$,R1         ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.
```

```
4946 025264  004767  172416                JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
4947 025270  103402                        BCS     12$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
4948 025272  052705  100002                 BIS     #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
4949 025276  005267  000042          12$:    INC     50$             ;INCREMENT THE REGISTER
4950 025302  005267  000036                 INC     50$             ; POINTER BY 2.
4951 025306  005302                        DEC     R2              ;COUNT THE REGISTER.
4952 025310  001351                        BNE     8$              ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
4953
4954                                ;*
4955                                ; DONE CHECKING DEVICE REGISTER ADDRESSES.
4956                                ;    REPORT ANY ERRORS AND EXIT.
4957                                ;-
4958
4959 025312  016767  155000  152464  40$:     MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4960 025320  005705                         TST     R5              ;CHECK THE ERROR FLAGS.
4961 025322  100012                        BPL     60$             ;EXIT ROUTINE IF NO ERRORS.
4962                                ;*
4963                                ; REPORT "DEVICE REGISTER ACCESS TEST FAILED"
4964                                ;-
4965 025324                                ERROR
     025324  104460                                                                  TRAP    C$ERROR
4966
4967
4968 025326                                DODU    UNITN           ;DROP THIS UNIT FROM FUTHER TESTING.
     025326  016700  154700                                                          MOV     UNITN,R0
     025332  104451                                                                  TRAP    C$DODU
4969 025334  005067  154742                 CLR     CTRLCF          ;INDICATE NO CTRL-C ABORT FROM TEST.
4970 025340                                DOCLN                   ;ABORT THIS SUB PASS.
     025340  104444                                                                  TRAP    C$DCLN
4971 025342  000402                        BR      60$             ;
4972
4973                                ;****************** LOCAL STORAGE. *********************************************
4974 025344  000000                 50$:    .WORD   0               ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
4975 025346  000000                 52$:    .WORD   0               ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
4976                                ;********************* END *****************************************************
4977
4978 025350  005067  154726          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
4979 025354                                ENDTST
     025354                                                                  L10026:
     025354  104401                                                                  TRAP    C$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.
                                ;*
                                ;-- ************************************************************************
025356                                  BGNTST
025356                                                                          T2::
        000002                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025356  012767  000002  154734          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (2)
025364  012767  177777  154710          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
025372                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
025372  012700  000240                                                          MOV     #PRI05,R0
025376  104441                                                                  TRAP    C$SPRI
025400  012767  000001  156354          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
025406  012767  005632  156352          MOV     #EM0201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
025414  012767  015776  156346          MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;-
025422  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
025426  012702  000040                  MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
025432  005003                          CLR     R3              ;WAITING FOR BIT TO CLEAR.
025434  016704  154576                  MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
025440  004767  172454                  JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
025444  103410                          BCS     2$              ;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.
                                ;-
025446                                  BRESET                  ;NO, TRY TO JOG DEVICE WITH BUS RESET.
025446  104433                                                                  TRAP    C$RESET
025450  004767  174624                  JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
025454  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
025460  004767  172434                  JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
025464  103016                          BCC     4$              ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
                                ;+
                                ; SET THE MASTER RESET BIT AND VERIFY THAT IT CLEARS WITHIN THE PROPER TIME.
                                ;-
025466  012701  011610          2$:     MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
025472  010214                          MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
025474  004767  172420                  JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
025500  103010                          BCC     4$              ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
025502  012702  011610                  MOV     #5000.,R2
025506  160102                          SUB     R1,R2           ;CALCULATE # OF MS FOR MR TO CLEAR.
025510  001413                          BEQ     6$              ;GO REPORT ERROR IF MR CLEAR IMMEDIATELY.
025512  020227  000764                  CMP     R2,#500.
025516  002417                          BLT     8$              ;GO REPORT ERROR IF MR CLEAR IN < 1/2 SECOND.
025520  000424                          BR      60$             ;EXIT THE TEST WITHOUT ERROR.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                   ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
025522  012767  000311  156234  4$:     MOV     #201.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
025530  012701  005665                  MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
025534                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #201 <<<<<
```

```
      025534  104460                                                          TRAP    C$ERROR
5034  025536  000415                    BR      60$             ;EXIT THE TEST.
5035
5036                                  ;REPORT MR BIT CLEAR IMMEDIATELY AFTER DUT RESET.
5037  025540  012767  000312  156216  6$:   MOV     #202.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
5038  025546  012701  006040            MOV     #EM0203,R1      ;SELECT ERROR MESSAGE.
5039  025552                            ERROR                   ;REPORT ERROR.          >>>>> ERROR #202 <<<<<
      025552  104460                                                          TRAP    C$ERROR
5040  025554  000406                    BR      60$             ;EXIT THE TEST.
5041
5042                                  ;REPORT MR CLEAR WITHIN 1/2 SECOND OF DUT RESET.
5043  025556  012767  000313  156200  8$:   MOV     #203.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
5044  025564  012701  006203            MOV     #EM0204,R1      ;SELECT ERROR MESSAGE.
5045  025570                            ERROR                   ;REPORT ERROR.          >>>>> ERROR #203 <<<<<
      025570  104460                                                          TRAP    C$ERROR
5046
5047  025572                    60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      025572  012700  000340                                                  MOV     #PRI07,R0
      025576  104441                                                          TRAP    C$SPRI
5048  025600  005067  154476            CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5049  025604                            ENDTST
      025604                                                          L10027:
      025604  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.
025606  012767  000003  154504     MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (3)
025614  012767  177777  154460     MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
025622                          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
025622  012700  000240                                                  MOV     #PRI05,R0
025626  104441                                                          TRAP    C$SPRI
025630  012767  000001  156124     MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
025636  012767  006362  156122     MOV     #EM0301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
025644  012767  015776  156116     MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;-
025652  012701  011610          10$:    MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
025656  012702  000040             MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
025662  005003                     CLR     R3              ;WAITING FOR BIT TO CLEAR.
025664  016704  154346             MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
025670  004767  172224             JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
025674  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.
025676  104433                                                          TRAP    C$RESET
025700  004767  174374             JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
025704  012701  011610             MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
025710  004767  172204             JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
025714  103024                     BCC     6$              ;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.
                                ;-
025716  012701  000310          2$:     MOV     #200.,R1        ;TIME-OUT VALUE IS 1/5 SECOND.
025722  010214                     MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
025724  004767  174350             JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
025730  004767  172164             JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
025734  103007                     BCC     4$              ;GO FIND OUT WHAT IS WRONG IF MR NOT CLEAR.
025736  012702  000310             MOV     #200.,R2
025742  160102                     SUB     R1,R2           ;CALCULATE # OF MS FOR MR TO CLEAR.
025744  020227  000012             CMP     R2,#10.
025750  002415                     BLT     8$              ;GO REPORT ERROR IF MR CLEAR IN < 10 MS.
025752  000431                     BR      60$             ;EXIT THE TEST WITHOUT ERROR.
                                ;+
                                ; MR DID NOT CLEAR WITHIN 1/5 SECOND, SEE IF IT CLEARS WITHIN 5 SECONDS.
                                ;-
025754  012701  011300          4$:     MOV     #4800.,R1       ;TIME-OUT VALUE IS 5 SECONDS MINUS 1/5 SECOND.
025760  004767  172134             JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
025764  103416                     BCS     10$             ;GO REPORT ERROR IF MR CLEARED FINALLY.
```

```
5104                                    ;*
5105                                    ; ERROR REPORTS:
5106                                    ;-
5107                                        ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5108 025766  012767  000455  155770  6$:       MOV     #0301.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5109 025774  012701  005665                   MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
5110 026000                                   ERROR                   ;REPORT ERROR.          >>>>> ERROR #0301 <<<<<
     026000  104460                                                                              TRAP    C$ERROR
5111 026002  000415                           BR      60$             ;EXIT THE TEST.
5112
5113                                        ;REPORT MR BIT CLEAR WITHIN 10 MS AFTER DUT RESET.
5114 026004  012767  000456  155752  8$:       MOV     #0302.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5115 026012  012701  006434                   MOV     #EM0302,R1      ;SELECT ERROR MESSAGE.
5116 026016                                   ERROR                   ;REPORT ERROR.          >>>>> ERROR #0302 <<<<<
     026016  104460                                                                              TRAP    C$ERROR
5117 026020  000406                           BR      60$             ;EXIT THE TEST.
5118
5119                                        ;REPORT MR CLEARED BETWEEN 1/5 SECOND AND 5 SECONDS OF DUT RESET.
5120 026022  012767  000457  155734  10$:      MOV     #0303.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5121 026030  012701  006574                   MOV     #EM0303,R1      ;SELECT ERROR MESSAGE.
5122 026034                                   ERROR                   ;REPORT ERROR.          >>>>> ERROR #0303 <<<<<
     026034  104460                                                                              TRAP    C$ERROR
5123
5124 026036                             60$:  SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     026036  012700  000340                                                                      MOV     #PRI07,R0
     026042  104441                                                                              TRAP    C$SPRI
5125 026044  005067  154232                    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5126 026050                                   ENDTST
     026050                                                                          L10030:
     026050  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.
                                ;*
                                ;-- ************************************************************************
026052                                  BGNTST
026052                                                                          T4::
026052                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
026052  012700  000240                                                          MOV     #PRI05,R0
026056  104441                                                                  TRAP    C$SPRI
        000004                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026060  012767  000004  154232           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (4)
026066  012767  177777  154206           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
026074  012767  000001  155660           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026102  012767  006753  155656           MOV     #EM0401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026110  012767  015776  155652           MOV     #ER0201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
026116  012701  011610                   MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
026122  012702  000040                   MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
026126  005003                           CLR     R3             ;WAITING FOR BIT TO CLEAR.
026130  016704  154102                   MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
026134  010214                           MOV     R2,(R4)        ;SET THE DUT MASTER RESET BIT.
026136  004767  174136                   JSR     PC,SKPSTS      ;SKIP THE SELFTEST.
026142  004767  171752                   JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026146  103015                           BCC     4$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; READ 6 CHARACTERS FROM THE DUT AND VERIFY THAT THEY ARE VALID SELFTEST
                                ;  CODES.
                                ;-
026150  012400                           MOV     (R4)+,R0       ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
026152  012701  000006                   MOV     #6,R1          ;INITIALIZE THE LOOP COUNTER.
026156  011402                  2$:      MOV     (R4),R2        ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
026160  010200                           MOV     R2,R0
026162  042700  177476                   BIC     #177476,R0     ;REMOVE ALL BUT BITS SPECIFIC TO SELFTEST CODE.
026166  020027  000201                   CMP     R0,#201        ;CHECK THAT BITS 0,6, AND 7 ARE CORRECT.
026172  001012                          BNE     6$             ;GO REPORT ERROR IF CODE IS NOT SELFTEST CODE.
026174  005301                           DEC     R1             ;COUNT THIS LOOP ITERATION.
026176  001367                          BNE     2$             ;LOOP IF NOT ALL LINES DONE.
026200  000415                          BR      60$            ;EXIT TEST, NO ERROR FOUND.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                   ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
026202  012767  000621  155554  4$:      MOV     #0401.,ERRNBR  ;SET THE ERROR NUMBER IN ERROR TABLE.
026210  012701  005665                   MOV     #EM0202,R1     ;SELECT ERROR MESSAGE.
026214                                  ERROR                  ;REPORT ERROR.          >>>>> ERROR #0401 <<<<<
026214  104460                                                                          TRAP    C$ERROR
026216  000406                          BR      60$            ;EXIT THE TEST.

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

```
5181 026220  012767  000622  155536  6$:     MOV     #0402.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5182 026226  012701  007031                  MOV     #EM0402,R1      ;SELECT ERROR MESSAGE.
5183 026232                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0402 <<<<<
     026232  104460                                                                              TRAP    C$ERROR
5184
5185 026234                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     026234  012700  000340                                                                      MOV     #PRI07,R0
     026240  104441                                                                              TRAP    C$SPRI
5186 026242  005067  154034                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5187 026246                                  ENDTST
     026246                                                                      L10031:
     026246  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.
                                ;*
                                ;-- ************************************************************************
026250                                  BGNTST
026250                                                                          T5::
026250                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
026250  012700  000240                                                        MOV     #PRI05,R0
026254  104441                                                                TRAP    C$SPRI
        000005                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026256  012767  000005  154034           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (5)
026264  012767  177777  154010           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
026272  012767  000001  155462           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026300  012767  007201  155460           MOV    #EM0501,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026306  012767  015776  155454           MOV    #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
026314  012701  011610                   MOV    #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
026320  012702  000040                   MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
026324  005003                           CLR    R3              ;WAITING FOR BIT TO CLEAR.
026326  016704  153704                   MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
026332  010214                           MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
026334  004767  173740                   JSR    PC,SKPSTS       ;SKIP THE SELFTEST.
026340  004767  171554                   JSR    PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026344  103013                           BCC    4$              ;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.
                                ;-
026346  012400                           MOV    (R4)+,R0        ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
026350  012701  000010                   MOV    #8.,R1          ;INITIALIZE THE LOOP COUNTER.
026354  011402                  2$:      MOV    (R4),R2         ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
026356  012700  070000                   MOV    #70000,R0
026362  040200                           BIC    R2,R0           ;CALCULATE BIT MAP OF CLEAR RX ERROR FLAGS.
026364  001012                           BNE    6$              ;GO REPORT ERROR IF NOT ALL RX ERROR FLAGS SET.
026366  005301                           DEC    R1              ;COUNT THIS LOOP ITERATION.
026370  001371                           BNE    2$              ;LOOP IF NOT ALL LINES DONE.
026372  000415                           BR     60$             ;EXIT TEST, NO ERROR FOUND.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
026374  012767  000765  155362  4$:       MOV    #0501.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026402  012701  005665                   MOV    #EM0202,R1      ;SELECT ERROR MESSAGE.
026406                                   ERROR                  ;REPORT ERROR.          >>>>> ERROR #0501 <<<<<
026406  104460                                                                          TRAP    C$ERROR
026410  000406                           BR     60$             ;EXIT THE TEST.
                                    ;REPORT ONE OR MORE RX ERROR FLAGS FOUND CLEAR WITH SELFTEST CODE.
026412  012767  000766  155344  6$:       MOV    #0502.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
```

```
 5242 026420  012701  007255                MOV     #EM0502,R1      ;SELECT ERROR MESSAGE.
 5243 026424                                ERROR                   ;REPORT ERROR.          >>>>> ERROR #0502 <<<<<
      026424  104460                                                                               TRAP    C$ERROR
 5244
 5245 026426                        60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      026426  012700  000340                                                                       MOV     #PRI07,R0
      026432  104441                                                                               TRAP    C$SPRI
 5246 026434  005067  153642                CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5247 026440                                ENDTST
      026440                                                                          L10032:
      026440  104401                                                                               TRAP    C$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.
                                ;*
                                ;-- ****************************************************************************
026442                                  BGNTST
026442                                                                          T6::
026442                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
026442  012700  000240                                                        MOV     #PRI05,R0
026446  104441                                                                TRAP    C$SPRI
000006                                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026450  012767  000006  153642           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (6)
026456  012767  177777  153616           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
026464  012767  000001  155270           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026472  012767  007611  155266           MOV    #EM0601,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026500  012767  015776  155262           MOV    #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
026506  012701  011610                   MOV    #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
026512  012702  000040                   MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
026516  005003                           CLR    R3              ;WAITING FOR BIT TO CLEAR.
026520  016704  153512                   MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
026524  010214                           MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
026526  004767  173546                   JSR    PC,SKPSTS       ;SKIP THE SELFTEST.
026532  004767  171362                   JSR    PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026536  103016                           BCC    4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; CHECK THAT THE RX DATA AVAILABLE BIT IS SET.
                                ;-
026540  032714  000200                    BIT    #BIT7,(R4)      ;TEST THE DUT RX.DATA.AVAIL BIT.
026544  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.
                                ;-
026546  012705  001130                    MOV    #600.,R5        ;ALLOW READING 600 CHARS BEFORE ERROR.
026552  010403                           MOV    R4,R3
026554  012300                           MOV    (R3)+,R0        ;CALCULATE THE RBUF ADDRESS.
026556  011300                   2$:     MOV    (R3),R0         ;READ A CHARACTER FROM THE RX FIFO.
026560  032714  000200                    BIT    #BIT7,(R4)      ;TEST THE DUT RX.DATA.AVAIL BIT.
026564  001427                           BEQ    60$             ;EXIT TEST WITHOUT ERROR IF RX.DATA.AVAIL CLR.
026566  005305                            DEC    R5              ;COUNT THE CHARACTER JUST READ.
026570  001372                           BNE    2$              ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
026572  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.
026574  012767  001131  155162   4$:     MOV    #0601.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026602  012701  005665                   MOV    #EM0202,R1      ;SELECT ERROR MESSAGE.
026606                                   ERROR                  ;REPORT ERROR.          >>>>> ERROR #0601 <<<<<
026606  104460                                                                          TRAP    C$ERROR
```

```
5302 026610  000415                        BR      60$             ;EXIT THE TEST.
5303
5304                                    ;REPORT THAT RX.DATA.AVAIL BIT WAS NOT SET AFTER A RESET COMPLETION.
5305 026612  012767  001132  155144  6$:   MOV     #0602.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5306 026620  012701  007653              MOV     #EM0602,R1      ;SELECT ERROR MESSAGE.
5307 026624                              ERROR                   ;REPORT ERROR.          >>>>> ERROR #0602 <<<<<
     026624  104460                                                                              TRAP    C$ERROR
5308 026626  000406                        BR      60$             ;EXIT THE TEST.
5309
5310                                    ;REPORT THAT RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO.
5311 026630  012767  001133  155126  8$:   MOV     #0603.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5312 026636  012701  010033              MOV     #EM0603,R1      ;SELECT ERROR MESSAGE.
5313 026642                              ERROR                   ;REPORT ERROR.          >>>>> ERROR #0603 <<<<<
     026642  104460                                                                              TRAP    C$ERROR
5314
5315 026644                          60$:  SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     026644  012700  000340                                                                      MOV     #PRI07,R0
     026650  104441                                                                              TRAP    C$SPRI
5316 026652  005067  153424              CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5317 026656                              ENDTST
     026656                                                                          L10033:
     026656  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.
                                                                                        MOV     #PRI05,R0
                                                                                        TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
                                         MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (7)
                                         MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
                                         MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
                                         MOV    #EM0701,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                         MOV    #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
                                         MOV    #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
                                         MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
                                         CLR    R3              ;WAITING FOR BIT TO CLEAR.
                                         MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
                                         MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
                                        JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
                                        JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
                                        BCC     4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; CHECK THAT THE RX DATA VALID BIT IS SET.
                                ;-
                                         MOV    (R4)+,R0        ;INCREMENT POINTER TO PNT TO DUT RBUF REG.
                                         TST    (R4)            ;TEST THE DUT RX.DATA.VALID BIT.
                                        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.
                                ;-
                                         MOV    #600.,R5        ;ALLOW READING 600 CHARS BEFORE ERROR.
                                2$:      MOV    (R4),R0         ;READ A CHARACTER FROM THE RX FIFO.
                                        BPL     60$             ;EXIT TEST WITHOUT ERROR IF BIT IS CLEAR.
                                         DEC    R5              ;COUNT THE CHARACTER JUST READ.
                                        BNE     2$              ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
                                         BR     8$              ;GO REPORT ERROR IF RX.DATA.VALID WOULDN'T CLR.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                   ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
                                4$:      MOV    #0701.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
                                         MOV    #EM0202,R1      ;SELECT ERROR MESSAGE.
                                        ERROR                   ;REPORT ERROR.          >>>>> ERROR #0701 <<<<<
                                                                                        TRAP    C$ERROR
                                        BR      60$             ;EXIT THE TEST.
```

```
5372                                        ;REPORT THAT RX.DATA.VALID BIT WAS NOT SET AFTER A RESET COMPLETION.
5373 027020  012767  001276  154736  6$:     MOV     #0702.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5374 027026  012701  010261                 MOV     #EM0702,R1      ;SELECT ERROR MESSAGE.
5375 027032                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0702 <<<<<
     027032  104460                                                                              TRAP    C$ERROR
5376 027034  000406                         BR      60$             ;EXIT THE TEST.
5377
5378                                        ;REPORT THAT RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO.
5379 027036  012767  001277  154720  8$:     MOV     #0703.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5380 027044  012701  010441                 MOV     #EM0703,R1      ;SELECT ERROR MESSAGE.
5381 027050                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0703 <<<<<
     027050  104460                                                                              TRAP    C$ERROR
5382
5383 027052                         60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     027052  012700  000340                                                                      MOV     #PRI07,R0
     027056  104441                                                                              TRAP    C$SPRI
5384 027060  005067  153216                 CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5385 027064                                 ENDTST
     027064                                                                         L10034:
     027064  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.
027066  012700  000240                                                          MOV     #PRI05,R0
027072  104441                                                                  TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027074  012767  000010  153216         MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (8)
027102  012767  177777  153172         MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
027110  012767  000001  154644         MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027116  012767  010624  154642         MOV     #EM0801,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
027124  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
027130  012702  000040                  MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
027134  005003                          CLR     R3              ;WAITING FOR BIT TO CLEAR.
027136  016704  153074                  MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
027142  010214                          MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
027144  004767  173130                 JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
027150  004767  170744                 JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
027154  103016                         BCC     4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; READ CHARACTERS FROM THE DUT RX FIFO AND VERIFY THAT THE LINE NUMBERS ARE
                                ;  CORRECT.
                                ; EIGHT CHARACTERS ARE READ FROM THE FIFO.
                                ;-
027156  005001                          CLR     R1              ;CLEAR THE LINE COUNTER.
027160  012400                          MOV     (R4)+,R0        ;INCREMENT POINTER TO PNT TO THE DUT RBUF REG.
027162  011402                 2$:      MOV     (R4),R2         ;READ A CHARACTER FROM THE DUT RX FIFO.
027164  010203                          MOV     R2,R3
027166  000303                          SWAB    R3
027170  042703  177760                 BIC     #177760,R3      ;REMOVE ALL BUT LINE NUMBER BITS.
027174  020301                         CMP     R3,R1           ;COMPARE WITH EXPECTED LINE NUMBER.
027176  001017                         BNE     6$              ;GO REPORT ERROR IF LINE NUMBERS DON'T MATCH.
027200  005201                          INC     R1              ;INCREMENT THE EXPECTED LINE NUMBER.
027202  020127  000010                 CMP     R1,#8.          ;COMPARE NUMBER OF CODES READ WITH MAX.
027206  001365                         BNE     2$              ;LOOP UNTIL CODES FOR ALL LINES ARE READ.
027210  000423                          BR      60$             ;EXIT TEST WITHOUT ERROR.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                   ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
027212  012767  001441  154544 4$:      MOV     #0801.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
027220  012767  016072  154542          MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
027226  012701  005665                  MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
027232                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0801 <<<<<
027232  104460                                                                          TRAP    C$ERROR
```

```
 5440 027234  000411                        BR      60$             ;EXIT THE TEST.
 5441
 5442                                ;REPORT THAT RX LINE NUMBER FIELD IS WRONG FOR SELFTEST CODE.
 5443 027236  012767  001442  154520  6$:    MOV     #0802.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
 5444 027244  012767  016072  154516         MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
 5445 027252  012701  010672                 MOV     #EM0802,R1      ;SELECT ERROR MESSAGE.
 5446 027256                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0802 <<<<<
      027256  104460                                                                            TRAP    C$ERROR
 5447
 5448 027260                          60$:   SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      027260  012700  000340                                                                    MOV     #PRI07,R0
      027264  104441                                                                            TRAP    C$SPRI
 5449 027266  005067  153010                 CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5450 027272                                 ENDTST
      027272                                                                        L10035:
      027272  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.
                                ;*
                                ;-- ******************************************************************************
027274                                  BGNTST
027274                                                                  T9::
027274                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
027274  012700  000240                                                          MOV     #PRI05,R0
027300  104441                                                                  TRAP    C$SPRI
                000011           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027302  012767  000011  153010           MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (9)
027310  012767  177777  152764           MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
027316  012767  000001  154436           MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027324  012767  001605  154432           MOV     #0901.,ERRNBR   ;SET THE ERROR NUMBER.
027332  012767  010742  154426           MOV     #EM0901,ERRMSG  ;SET THE ERROR MESSAGE
                                ;+
                                ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                ;-
027340  012701  005670                   MOV     #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
027344  012702  000040                   MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
027350  005003                           CLR     R3              ;WAITING FOR BIT TO CLEAR.
027352  016704  152660                   MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
027356  004767  170536                   JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
027362  103027                           BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
                                ;+
                                ; RESET THE DUT, SKIP THE SELF-TEST.
                                ;-
                                         MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
027364  010214                           JSR     PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
027366  004767  172706
                                ;+
                                ; WAIT FOR MASTER RESET TO CLEAR. DELAY FOR 500 MILLI-SECS BEFORE PURGING
                                ; THE FIFO.
                                ;-
027372  012704  000764                   MOV     #500.,R4        ;TIME-OUT VALUE IS 500 MILLI-SECONDS.
027376  004767  170456                   JSR     PC,DELAY        ;WAIT FOR BMP TO BEGIN EXECUTION.
027402  004767  171070                   JSR     PC,PUFIFO       ;PURGE THE FIFO, SAVING ANY BMP CODES.
027406  103015                           BCC     50$             ;ABORT THE TEST IF THE FIFO DID NOT CLEAR.
                                ;+
                                ; REPORT THE ERROR IF ANY BMP CODES WERE FOUND.
                                ;-
027410  016702  153104                   MOV     BMPCQP,R2       ;GET THE CONTENTS OF THE POINTER TO THE BMP Q.
027414  012703  002522                   MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
027420  020203                           CMP     R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
027422  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"
```

F12


```
5506 027424  012701  011002                ERRDF   #EM0902,R1      ;PASS THE MESSAGE TO BE REPORTED.
5507 027430                                        0901,EM0901,ER9301 ;            >>>>> ERROR #0901 <<<<<.
     027430  104455                                                                 TRAP    C$ERDF
     027432  001605                                                                 .WORD   901
     027434  010742                                                                 .WORD   EM0901
     027436  017216                                                                 .WORD   ER9301
5508 027440  000405                BR      60$
5509
5510 027442  012767  001606  154314  50$:    MOV     #902.,ERRNBR    ;SET >>>>> ERROR #0902 <<<<<.
5511 027450  004767  172752                JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
5512
5513 027454                        60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     027454  012700  000340                                                        MOV     #PRI07,R0
     027460  104441                                                                 TRAP    C$SPRI
5514 027462  005067  152614                CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5515 027466                                ENDTST
     027466                                                                 L10036:
     027466  104401                                                                 TRAP    C$ETST
```

```
5517
5518
5519                                    .SBTTL  HARDWARE TEST           - SKSELF -
5520                                    ;++ ************************************************************************
5521                                    ;*              - SKIP SELF-TEST TEST -
5522                                    ;*      THIS TEST VERIFIES THAT THE DUT SKIPS THE SELF-TEST WITHIN THE
5523                                    ;*      TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS
5524                                    ;*      COMPLETION.
5525                                    ;*
5526                                    ;-- ************************************************************************
5527 027470                                        BGNTST
         027470                                                                          T10::
5528 027470                                        SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
         027470  012700  000240                                                         MOV     #PRI05,R0
         027474  104441                                                                 TRAP    C$SPRI
5529             000012                             TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5530 027476  012767  000012  152614                 MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (10)
5531 027504  012767  177777  152570                 MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
5532 027512  012767  000001  154242                 MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5533 027520  012767  011036  154240                 MOV     #EM1001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5534 027526  012767  016072  154234                 MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5535                                    ;+
5536                                    ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5537                                    ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5538                                    ;-
5539 027534  012701  011610                        MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
5540 027540  012702  000040                        MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
5541 027544  005003                                CLR     R3              ;WAITING FOR BIT TO CLEAR.
5542 027546  016704  152464                        MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
5543 027552  004767  170342                        JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5544 027556  103037                                BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
5545                                    ;+
5546                                    ; DETERMINE IF THE DUT TAKES TOO SHORT OR TOO LONG A TIME TO SKIP THE SELF-TEST
5547                                    ;  SET-UP A TIME-OUT OF 50 MILLI-SECOND, IF MR IS CLEAR IN LESS THAN 10 MILLI
5548                                    ;    -SECOND, OR GREATER THAN 50 MILLI-SECONDS, REPORT THE ERROR.
5549                                    ;-
5550 027560  012701  000062                        MOV     #50.,R1         ;TIME-OUT VALUE IS 50 MILLI-SECONDS.
5551 027564  010214                                MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
5552 027566  004767  172506                        JSR     PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
5553 027572  004767  170322                        JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5554 027576  103011                                BCC     2$              ;GO REPORT ERR IF SKIPPING STEST TOOK TOO LONG.
5555 027600  020127  000050                        CMP     R1,#40.
5556 027604  003015                                BGT     4$              ;GO REP ERR IF SELFTEST COMPLETED IN < 10 MS.
5557                                    ;+
5558                                    ; SELF-TEST COMPLETED WITHIN 10 MILLI-SEC TO 50 MILLI-SECONDS.
5559                                    ;  VERIFY THAT THE SELF-TEST CODES IN THE FIFO ARE "GOOD" CODES ,IE THE DUT
5560                                    ;  SUCCESSFULLY COMPLETED THE SELF-TEST.
5561                                    ; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 1003 THRU 1007 <<<<<.
5562                                    ;-
5563 027606  012767  001753  154150                 MOV     #1003.,ERRNBR   ;SET ERROR NUMBER TO 1003.
5564 027614  004767  171630                        JSR     PC,RSTRPT       ;CHECK SELF-TEST CODES IN THE FIFO.
5565 027620  000423                                BR      60$             ;EXIT TEST.
5566                                    ;+
5567                                    ; ERROR REPORTS:
5568                                    ;-
5569                                       ;REPORT SKIP SELF-TEST TOOK TOO LONG.
5570 027622  012767  001751  154134  2$:           MOV     #1001.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
```

```
 5571 027630  012701  011071                    MOV     #EM1002,R1      ;SELECT ERROR MESSAGE.
 5572 027634                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #1001 <<<<<
      027634  104460                                                                                  TRAP    C$ERROR
 5573 027636  000414                            BR      60$             ;EXIT THE TEST.
 5574
 5575                                   ;REPORT SKIP SELF-TEST COMPLETED TOO SOON.
 5576 027640  012767  001752  154116  4$:       MOV     #1002.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
 5577 027646  012701  011156                    MOV     #EM1003,R1      ;SELECT ERROR MESSAGE.
 5578 027652                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #1002 <<<<<
      027652  104460                                                                                  TRAP    C$ERROR
 5579 027654  000405                            BR      60$             ;EXIT THE TEST.
 5580
 5581 027656  012767  001753  154100  50$:      MOV     #1003.,ERRNBR   ;SET ERROR NUMBER.
 5582 027664  004767  172536                    JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
 5583
 5584 027670                            60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      027670  012700  000340                                                                          MOV     #PRI07,R0
      027674  104441                                                                                  TRAP    C$SPRI
 5585 027676  005067  152400                    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5586 027702                                    ENDTST
      027702                                                                    L10037:
      027702  104401                                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - DFSKST -
                                ;++ ************************************************************************
                                ;*              - DIAGNOSTIC FAIL BIT, SKIP SELF-TEST TEST -
                                ;*      THIS TEST VERIFIES THAT THE DIAGNOSTIC FAIL BIT OF THE DUT, CORRECTLY
                                ;*      CHANGES STATE AS THE ON-BOARED SELFTEST IS SKIPPED.
                                ;*
                                ;-- ************************************************************************
027704                                  BGNTST
027704                                                                          T11::
027704                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
027704  012700  000240                                                        MOV     #PRI05,R0
027710  104441                                                                TRAP    C$SPRI
        000013                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027712  012767  000013  152400          MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (11)
027720  012767  177777  152354          MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
027726  012767  000001  154026          MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027734  012767  011234  154024          MOV     #EM1101,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
027742  012767  016072  154020          MOV     #ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                ;-
027750  012701  011610                    MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
027754  012702  000040                    MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
027760  005003                           CLR     R3             ;WAITING FOR BIT TO CLEAR.
027762  016704  152250                   MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
027766  004767  170126                   JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
027772  103044                           BCC     50$            ;ABORT THE TEST IF MR DID NOT CLEAR.
                                ;+
                                ; RESET THE DUT, SKIP THE SELF-TEST.
                                ;-
027774  010214                           MOV     R2,(R4)        ;SET THE DUT MASTER RESET BIT.
027776  004767  172276                   JSR     PC,SKPSTS      ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
                                ;+
                                ; SET TIME OUT OF 5 MILLI SECONDS, WAIT FOR DIAG_FAIL BIT TO SET.
                                ;  IF TIME-OUT OCCURS GO REPORT THE ERROR.
                                ;-
030002  012701  000005                    MOV     #5,R1          ;TIME-OUT VALUE IS 5 MILLI-SECONDS.
030006  012702  020000                    MOV     #BIT13,R2      ;WAITING FOR DIAGNOSTIC FAIL BIT.
030012  010203                           MOV     R2,R3          ;WAITING FOR BIT TO SET.
030014  016704  152216                   MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
030020  004767  170074                   JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
030024  103020                           BCC     4$             ;IF DIAG_FAIL DID NOT SET, GO REPORT ERROR.
                                ;+
                                ; SET TIME-OUT OF 15 MILLI-SECS, WAIT FOR DIAG_FAIL TO CLEAR.
                                ;  IF TIME-OUT OCCURS GO REPORT THE ERROR.
                                ;  VERIFY THE DIAG_FAIL BIT IS IN A STABLE STATE BEFORE CONTINUING.  LOOP
                                ;  BACK IF THE STATE WAS TRANSITORY, USING THE REMAINDER OF THE 15 MS TIME-OUT.
                                ;-
030026  012701  000017                    MOV     #15.,R1        ;TIME-OUT VALUE IS 15 MILLI-SECONDS.
030032  005003                  2$:       CLR     R3             ;WAITING FOR BIT TO CLEAR.
030034  004767  170060                   JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
030040  103012                           BCC     4$             ;IF DIAG_FAIL DID NOT CLEAR, GO REPORT ERROR.
030042  010105                           MOV     R1,R5          ;SAVE THE REMAINING TIME-OUT VALUE.
```

```
 5642 030044  012701  000001                MOV     #1,R1           ;SET TIME-OUT OF 1 MILLI-SECOND.
 5643 030050  052703  020000                BIS     #BIT13,R3       ;WAIT FOR BIT TO SET.
 5644 030054  004767  170040                JSR     PC,MSLGET       ;DOUBLE CHECK TO ELIMINATE NOISE PROBLEMS.
 5645 030060  103016                        BCC     60$             ;EXIT IF DIAG_FAIL BIT STILL CLEAR.
 5646 030062  010501                        MOV     R5,R1           ;PASS THE REMAINING TIME-OUT VALUE.
 5647 030064  000762                        BR      2$              ;LOOP TO CHECK AGAIN.
 5648                                ;+
 5649                                ; ERROR REPORTS:
 5650                                ;-
 5651                                   ;REPORT DIAGNOSTIC FAIL BIT BAD.
 5652 030066  012767  002115  153670  4$:   MOV     #1101.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
 5653 030074  012701  011533                MOV     #EM1205,R1      ;SELECT ERROR MESSAGE.
 5654 030100                                ERROR                   ;REPORT ERROR.          >>>>> ERROR #1101 <<<<<
      030100  104460                                                                              TRAP    C$ERROR
 5655 030102  000405                        BR      60$             ;EXIT THE TEST.
 5656
 5657 030104  012767  002116  153652  50$:  MOV     #1102.,ERRNBR   ;SET THE ERROR NUMBER FOR TSABRT RTN.
 5658 030112  004767  172310                JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
 5659
 5660 030116                        60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      030116  012700  000340                                                          MOV     #PRI07,R0
      030122  104441                                                                  TRAP    C$SPRI
 5661 030124  005067  152152                CLR     CTRLCF          ;INDICATE THAT WE COMPLETED A TEST.
 5662 030130                                ENDTST
      030130                                                                  L10040:
      030130  104401                                                                  TRAP    C$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.
                                ;*
                                ;-- ************************************************************************
030132                                  BGNTST
030132                                                                          T12::
030132                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
030132  012700  000240                                                          MOV     #PRI05,R0
030136  104441                                                                  TRAP    C$SPRI
        000014                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
030140  012767  000014  152152           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (12)
030146  012767  177777  152126           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
030154  012767  000001  153600           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
030162  012767  011307  153576           MOV    #EM1201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
030170  012767  016072  153572           MOV    #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                ;-
030176  012701  011610                    MOV    #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
030202  012702  000040                    MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
030206  005003                            CLR    R3              ;WAITING FOR BIT TO CLEAR.
030210  016704  152022                    MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
030214  004767  167700                   JSR    PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030220  103067                            BCC    50$             ;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 5 SECOND, IF MR IS CLEAR IN LESS THAN 1/2 SECOND, OR
                                ;  GREATER THAN 5 SECONDS, REPORT THE ERROR.
                                ;-
030222  012701  011610                    MOV    #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
030226  010214                            MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
030230  004767  167664                   JSR    PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030234  103034                            BCC    4$              ;GO REPORT ERROR SELFTEST TOOK TOO LONG.
030236  012702  011610                    MOV    #5000.,R2
030242  160102                            SUB    R1,R2           ;CALCULATE # OF MS SELFTEST TO COMPLETE.
030244  020227  000062                    CMP    R2,#50.
030250  002435                            BLT    6$              ;SELFTEST SKIPPED?  YES, GO REPORT ERROR.
030252  020227  000764                    CMP    R2,#500.
030256  002441                            BLT    8$              ;GO REP ERR IF SELFTEST COMPLETED IN < 1/2 SEC.
                                ;+
                                ; SELF-TEST COMPLETED WITHIN 1SEC TO 5 SECONDS.
                                ;  CHECK THE STATE OF THE DIAGNOSTIC FAIL BIT, REPORT ERROR IF IT IS SET.
                                ;-
030260  032714  020000                    BIT    #BIT13,(R4)     ;DETERMINE IF THE DIAG_FAIL BIT IS CLEAR.
030264  001412                            BEQ    2$              ;SKIP ERROR REPORT IF BIT IS CLEAR.
                                ;REPORT DIAGNOSTIC FAIL BIT BAD.
030266  012767  002264  153470           MOV    #1204.,ERRNBR   ;SET ERROR NUMBER TO IN ERROR TABLE.
030274  012701  011533                    MOV    #EM1205,R1      ;SELECT THE ERROR MESSAGE.
030300                                   ERROR                  ;                >>>>> ERROR #1204 <<<<<
030300  104460                                                                  TRAP    C$ERROR
                                ;+
                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
```

```
                                        ;-
030302  032767  000100  151710            BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
030310  001440                            BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                                  ;DURING THE SOFTWARE QUESTIONS.
                                ;+
                                ; 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 <<<<<.
                                ;-
030312  012767  002265  153444  2$:       MOV     #1205.,ERRNBR   ;SET ERROR NUMBER TO 1205.
030320  004767  171124                    JSR     PC,RSTRPT       ;CHECK SELF-TEST CODES IN THE FIFO.
030324  000432                            BR      60$             ;EXIT TEST.
                                ;+
                                ; ERROR REPORTS:
                                ;-
                                    ;REPORT SELF-TEST TOOK TOO LONG TO COMPLETE.
030326  012767  002261  153430  4$:       MOV     #1201.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
030334  012701  011335                    MOV     #EM1202,R1      ;SELECT ERROR MESSAGE.
030340                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #1201 <<<<<
030340  104460                                                                                    TRAP    C$ERROR
030342  000423                            BR      60$             ;EXIT THE TEST.
                                    ;REPORT SELF-TEST DID NOT EXECUTE AFTER DUT RESET.
030344  012767  002262  153412  6$:       MOV     #1202.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
030352  012701  011477                    MOV     #EM1204,R1      ;SELECT ERROR MESSAGE.
030356                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #1202 <<<<<
030356  104460                                                                                    TRAP    C$ERROR
030360  000414                            BR      60$             ;EXIT THE TEST.
                                    ;REPORT SELF-TEST COMPETED TOO SOON.
030362  012767  002263  153374  8$:       MOV     #1203.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
030370  012701  011421                    MOV     #EM1203,R1      ;SELECT ERROR MESSAGE.
030374                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #1203 <<<<<
030374  104460                                                                                    TRAP    C$ERROR
030376  000405                            BR      60$             ;EXIT THE TEST.
030400  012767  002272  153356  50$:      MOV     #1210.,ERRNBR   ;SET THE ERROR NUMBER FOR TSABRT RTN.
030406  004767  172014                    JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
030412                          60$:      SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
030412  012700  000340                                                                    MOV     #PRI07,R0
030416  104441                                                                           TRAP    C$SPRI
030420  005067  151656                    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
030424                                    ENDTST
030424                                                                          L10041:
030424  104401                                                                           TRAP    C$ETST
```

```
5757
5758                                    .SBTTL  HARDWARE TEST           - STFAIL -
5759                                    ;** ************************************************************************************
5760                                    ;*              - SELF-TEST FAIL TEST -
5761                                    ;*      THIS TEST VERIFIES THAT THE DUT WILL REPORT SELFTEST ERRORS VIA THE
5762                                    ;*      FIFO.  AND THAT THE DIAGNOSTIC FAIL BIT WILL INDICATE THE ERROR.
5763                                    ;*      THIS IS ACCOMPLISHED VIA A SOFTWARE 'HOOK' IN THE SELF-TEST, WHICH
5764                                    ;*      FORCES A "PROC1 TO RAM ERROR" TO BE PLACED IN THE FIFO.
5765                                    ;*
5766                                    ;-- ************************************************************************************
5767 030426                                        BGNTST
      030426                                                                                    T13::
5768 030426                                        SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
      030426  012700  000240                                                                    MOV     #PRIO5,R0
      030432  104441                                                                            TRAP    C$SPRI
5769          000015                            TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5770 030434  012767  000015  151656             MOV     #TNUM,TSTNUM      ;SET UP THE TEST NUMBER.        (13)
5771 030442  012767  177777  151632             MOV     #-1,CTRLCF        ;INDICATE THAT WE ARE WITHIN A TEST.
5772 030450  012767  000001  153304             MOV     #1,ERRTYP         ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5773 030456  012767  011557  153302             MOV     #EM1301,ERRMSG    ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5774 030464  012767  016072  153276             MOV     #ER0503,ERRBLK    ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5775 030472  012767  002425  153264             MOV     #1301.,ERRNBR     ;SET ERROR NUMBER TO 1301.
5776                                    ;+
5777                                    ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5778                                    ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5779                                    ;-
5780 030500  012701  011610                        MOV     #5000.,R1         ;TIME-OUT VALUE IS 5.0 SECONDS.
5781 030504  012702  000040                        MOV     #BIT05,R2         ;WAITING FOR MASTER RESET BIT.
5782 030510  005003                                CLR     R3                ;WAITING FOR BIT TO CLEAR.
5783 030512  016704  151520                        MOV     CSRA,R4           ;BIT IS IN THE DUT'S CSR.
5784 030516  004767  167376                       JSR     PC,MSLGET         ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5785 030522  103064                               BCC     50$               ;GO REPORT ERROR IF MR DID NOT CLEAR.
5786
5787                                    ;+
5788                                    ; RESET THE DUT, DELAY FOR 25 MILLI-SECONDS BEFORE WRITING THE FAIL_SELF_TEST
5789                                    ; CODE TO TBUFFCT REGISTER ON CHANNEL 0.
5790                                    ;-
5791 030524  012777  000040  151504               MOV     #BIT05,@CSRA      ;SET DUT MASTER RESET BIT, SELECT CHANNEL 0.
5792 030532  012704  000031                        MOV     #25.,R4           ;PASS DELAY PERIOD OF 25 MILLI SECS.
5793 030536  004767  167316                       JSR     PC,DELAY          ;WAIT FOR SELFTEST TO INITIALISE.
5794 030542  012777  146314  151504               MOV     #146314,@TXBFCA   ;WRITE THE FAIL SELF-TEST CODE TO TBUFFCT REG.
5795                                    ;+
5796                                    ; WAIT UP TO 5 SECONDS FOR THE SELF-TEST TO COMPLETE.
5797                                    ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5798                                    ;-
5799 030550  005267  153210                        INC     ERRNBR            ;SET ERROR NUMBER TO 1302.
5800 030554  012701  011610                        MOV     #5000.,R1         ;TIME-OUT VALUE IS 5.0 SECONDS.
5801 030560  012702  000040                        MOV     #BIT05,R2         ;PASS THE BIT MAP OF THE BIT TO TEST.
5802 030564  005003                                CLR     R3                ;SET UP THE EXPECTED STATE.
5803 030566  016704  151444                        MOV     CSRA,R4           ;BIT IS IN THE DUT'S CSR.
5804 030572  004767  167322                       JSR     PC,MSLGET         ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5805 030576  103036                               BCC     50$               ;GO REPORT ERROR IF MR DID NOT CLEAR.
5806                                    ;+
5807                                    ; VERIFY THE DIAGNOSTIC FAIL BIT IS SET, INDICATING THE ERROR.
5808                                    ; REPORT ERROR IF DIAGNOSTIC FAIL BIT IS CLEAR.
5809                                    ;-
5810 030600  005267  153160                        INC     ERRNBR            ;SET ERROR NUMBER TO 1303.
```

```
5811 030604  032714  020000                  BIT     #BIT13,(R4)     ;CHECK THE STATE OF THE DIAG_FAIL BIT.
5812 030610  001425                        BEQ     10$             ;GO REPORT ERROR IF DIAG_FAIL BIT CLEAR.
5813                                    ;+
5814                                    ; REMOVE THE 8 SELF-TEST CODES FORM THE FIFO, AND VERIFY THAT AT LEAST
5815                                    ; ONE IS A PROC1 TO RAM ERROR CODE (231).
5816                                    ;-
5817 030612  005267  153146                  INC     ERRNBR          ;SET ERROR NUMBER TO 1304.
5818 030616  012700  000010                  MOV     #8.,R0          ;SET MAXIMUM READ COUNT.
5819 030622  005001                          CLR     R1              ;CLEAR THE CORRECT CODE COUNTER.
5820 030624  016704  151410                  MOV     RBUFA,R4        ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
5821 030630  011402                6$:       MOV     (R4),R2         ;READ A CODE FROM THE FIFO.
5822 030632  100020                        BPL     50$             ;GO REPORT ERROR IF THE FIFO IS EMPTY.
5823 030634  042702  007400                  BIC     #7400,R2        ;REMOVE THE LINE NUMBER FROM THE CODE.
5824 030640  120227  170231                  CMPB    R2,#170231      ;IS IT THE CORRECT ERROR CODE?.
5825 030644  001001                        BNE     8$              ;SKIP NEXT INSTRUCTION, IF NOT A 231 CODE.
5826 030646  005201                          INC     R1              ;INCREMENT COUNTER.
5827 030650  005300                8$:       DEC     R0              ;DECREMENT MAX READ COUNTER.
5828 030652  001366                        BNE     6$              ;LOOP IF 8 CODES HAVE NOT BEEN READ.
5829 030654  005701                          TST     R1              ;WERE ANY 231 CODES FOUND?.
5830 030656  001010                        BNE     60$             ;YES, THEN EXIT.
5831 030660  005267  153100                  INC     ERRNBR          ;NO, SET ERROR NUMBER TO 1305 AND REPORT ERROR.
5832                                ;REPORT SELF-TEST ERROR REPORTING BAD.
5833 030664  012701  011612        10$:      MOV     #EM1302,R1      ;SELECT ERROR MESSAGE.
5834 030670                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR <<<<<
     030670  104460                                                                               TRAP    C$ERROR
5835 030672  000402                        BR      60$             ;EXIT THE TEST.
5836
5837 030674  004767  171526        50$:      JSR     PC,TSABRT       ;REPORT NON-RELATED TEST ERROR.
5838
5839 030700                        60$:      SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     030700  012700  000340                                                                       MOV     #PRI07,R0
     030704  104441                                                                               TRAP    C$SPRI
5840 030706  005067  151370                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5841 030712                                 ENDTST
     030712                                                                     L10042:
     030712  104401                                                                               TRAP    C$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.
                                ;*
                                ;-- ************************************************************************
030714                                  BGNTST
030714                                                                  T14::
030714                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
030714  012700  000240                                                  MOV     #PRI05,R0
030720  104441                                                          TRAP    C$SPRI
                                        TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
000016                                  MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (14)
030722  012767  000016  151370          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
030730  012767  177777  151344          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
030736  012767  000001  153016          MOV     #EM1401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
030744  012767  011651  153014          MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
030752  012767  016072  153010
                                ;+
                                ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                ;-
030760  012701  005670                  MOV     #3000.,R1       ;TIME-OUT VALUE IS 3.0 SECONDS.
030764  012702  000040                  MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
030770  005003                          CLR     R3              ;WAITING FOR BIT TO CLEAR.
030772  016704  151240                  MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
030776  004767  167116                  JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
031002  103131                          BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
                                ;+
                                ; SET THE MASTER RESET BIT, AND SKIP THE SELF TEST.
                                ;-
031004  010214                          MOV     R2,(R4)         ;SET THE MASTER RESET BIT.
031006  004767  171266                  JSR     PC,SKPSTS       ;SKIP THE SELF TEST.
031012  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
031016  004767  167076                  JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
031022  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.
                                ;-
031024  012705  000040                  MOV     #32.,R5         ;SET MAXIMUM TRY COUNTER.
031030  012703  000143                  MOV     #99.,R3         ;SET AN INVALID ROM VERSION NUMBER FOR PROC_1.
031034  010304                          MOV     R3,R4           ;SET AN INVALID ROM VERSION NUMBER FOR PROC_2.
031036  012767  002571  152720          MOV     #1401.,ERRNBR   ;SET THE ERROR NUMBER TO 1401.
031044  012701  011710                  MOV     #EM1402,R1      ;SELECT MESSAGE TO BE REPORTED IF FIFO EMPTY.
031050  017702  151164          2$:     MOV     @RBUFA,R2       ;READ THE NEXT CHAR FROM THE FIFO.
031054  100077                          BPL     12$             ;GO REPORT ERROR IF FIFO EMPTY.
                                  ;+
                                  ; CHECK IF THE READ DATA IS A BMP CODE.
```

```
                                ;-
                                        MOV     #301,R0         ;SET-UP A BIT MASK OF A BMP CODE.
                                        BIC     R2,R0           ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
                                        BNE     4$              ;BRANCH IF NOT A BMP CODE.
                                        JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
                                        BR      8$              ;
                                ;+
                                ; CHECK IF THE READ DATA IS A SELF-TEST CODE.
                                ;-
                                4$:     MOV     #201,R0         ;SET-UP A BIT MASK OF A SELFTEST CODE.
                                        BIC     R2,R0           ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
                                        BEQ     8$              ;BRANCH IF IT IS A SELFTEST CODE.

                                ;+
                                ; THE READ DATA IS A ROM VERSION NUMBER, DETERMINE WHICH ONE IT IS.
                                ;-
                                        BIT     #BIT1,R2        ;CHECK THE PROCESSOR NUMBER BIT IN THE CODE.
                                        BEQ     6$              ;BRANCH IF IT IS PROC_1 ROM VERSION NUMBER.
                                        MOV     R2,R4           ;SAVE PROC_2 ROM VERSION NUMBER.
                                        BIC     #177603,R4      ;CLEAR ANY UNWANTED BITS.
                                        CLC                     ;CLEAR THE CARRY BIT.
                                        ROR     R4              ;SHIFT THE CODES ALONG TO GET THE ROM
                                        ROR     R4              ; VERSION NUMBER IN THE LOW 5 BITS.
                                        BR      8$              ;
                                6$:     MOV     R2,R3           ;SAVE PROC_1 ROM VERSION NUMBER.
                                        BIC     #177603,R3      ;CLEAR ANY UNWANTED BITS.
                                        CLC                     ;CLEAR THE CARRY BIT.
                                        ROR     R3              ;SHIFT THE CODE ALONG TO GET THE ROM
                                        ROR     R3              ; VERSION NUMBER IN THE LOW 5 BITS.
                                        CMP     R4,#99.         ;CHECK IF WE HAVE RECEIVE PROC_2 ROM CODE.
                                        BNE     10$             ;GO REPORT BOTH ROM VERSION NUMBERS.
                                ;+
                                ; RECEIVED ROM VERSION NUMBERS OUT OF SEQUENCE.
                                ;  IE, PROC_1'S ROM VERSION NUMBER FOUND IN THE FIFO BEFORE PROC_2'S.
                                ;-
                                        MOV     #EM1403,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
                                        MOV     #1402.,ERRNBR   ;SET THE ERROR NUMBER.
                                        BR      12$             ;GO REPORT ERROR.

                                8$:     DEC     R5              ;DECREMENT THE MAX TRY COUNTER.
                                        BNE     2$              ;LOOP TO GET THE NEXT CHAR FROM THE FIFO.
                                        MOV     #EM1404,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
                                        MOV     #1403.,ERRNBR   ;SET THE ERROR NUMBER.
                                        BR      12$             ;GIVE UP, GO REPORT ERROR.
                                ;+
                                ; IF THIS IS THE FIRST PASS, AND SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
                                ; THEN REPORT THE ROM VERSION NUMBERS TO THE OPERATOR.
                                ;-
                                10$:    BIT     #BIT0,OPTION    ;CHECK ON THE STATE OF THE SOFTWARE SWITCH.
                                        BEQ     60$             ;EXIT IF NO ROM VERSION PRINTOUT WAS REQUESTED.
                                        CMP     PASCNT,#1       ;CHECK IF THIS IS THE FIRST PASS.
                                        BGT     60$             ;EXIT IF ROM VERS HAVE ALREADY BEEN REPORTED.
                                        PRINTB  #EF1401,R3,R4   ;PRINT THE ROM VERSION NUMBERS.
                                                                                MOV     R4,-(SP)
                                                                                MOV     R3,-(SP)
                                                                                MOV     #EF1401,-(SP)
                                                                                MOV     #3,-(SP)
```

```
        031242  010600                                                          MOV     SP,R0
        031244  104414                                                          TRAP    C$PNTB
        031246  062706  000010                                                  ADD     #10,SP
5950    031252  000412                  BR      60$             ;EXIT THIS TEST.
5951                                    ;-
5952                                    ; ERROR REPORTS:
5953                                    ;-
5954    031254  012767  016210  152506  12$:    MOV     #ER1401,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
5955    031262                          ERROR                           ;REPORT ERROR.          >>>>> ERROR <<<<<
        031262  104460                                                          TRAP    C$ERROR
5956    031264  000405                  BR      60$
5957
5958    031266  012767  002575  152470  50$:    MOV     #1405.,ERRNBR   ;SET UP ERROR NUMBER FOR TSABRT RTN.
5959    031274  004767  171126          JSR     PC,TSABRT               ;REPORT NON-TEST RELATED ERROR.
5960
5961    031300                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
        031300  012700  000340                                                  MOV     #PRI07,R0
        031304  104441                                                          TRAP    C$SPRI
5962    031306  005067  150770          CLR     CTRLCF                  ;INDICATE THAT WE COMPLETED THE TEST.
5963    031312                          ENDTST
        031312                                                          L10043:
        031312  104401                                                          TRAP    C$ETST
```

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

                                        BGNTST
                                                                                        T15::
                                         SETPRI #PRI05          ;ALLOW THE LTC TO INTERRUPT.
031314  012700  000240                                                          MOV     #PRI05,R0
031320  104441                                                                  TRAP    C$SPRI
        000017                  TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031322  012767  000017  150770 MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (16)
031330  012767  177777  150744 MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
031336  012767  000001  152416 MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
031344  012767  003101  152412 MOV     #1601.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
031352  012767  012254  152406 MOV     #EM1604,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
031360  005067  151072         CLR     ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
031364  012700  002460         MOV     #ERCNTB,R0
031370  004767  166364        JSR     PC,CLR16W       ;CLEAR THE ERROR COUNTER TABLE.
031374  005067  150700         CLR     EXOERR          ;CLEAR THE "EXIT ON ERROR" FLAG
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1601 <<<<<.
                                ;-
031400  004767  167706                JSR     PC,RESETT       ;RESET THE DMU-11, REPORT ANY ERRORS FOUND.
031404  103402                        BCS     .+6             ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
031406  000167  000142                JMP     60$             ;YES,  EXIT THE TEST.
                                ;+
                                ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR
                                ;-
031412  005267  152346                 INC     ERRNBR          ;SET THE ERROR REPORT NUMBER TO 1602.
031416  012702  000017                 MOV     #17,R2          ;SET LOOP COUNT.
031422  016704  150610                 MOV     CSRA,R4         ;GET CSR ADDRESS.
031426  010214                2$:      MOV     R2,(R4)         ;WRITE COUNT TO CSR.
031430  011401                         MOV     (R4),R1         ;READ BACK THE CONTENTS OF THE CSR
031432  042701  177760                 BIC     #177760,R1      ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
031436  020102                         CMP     R1,R2           ;CHECK FOR CORRECT DATA WRITTEN/READ.
031440  001412                        BEQ     4$              ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                    ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
031442  012767  016352  152320         MOV     #ER1601,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
031450  005003                         CLR     R3              ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
031452  005005                         CLR     R5              ;CAUSE REPORT OF LINE 0.
031454                               ERROR                   ;  >>>>> ERROR # 1602 <<<<<
031454  104460                                                                  TRAP    C$ERROR
                                ;+
                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
                                ;-
031456  032767  000100  150534         BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR BEEN REQUESTED ?
031464  001433                        BEQ     60$             ;EXIT THE TEST IF IT HASN'T.

031466  005302                4$:      DEC     R2              ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
031470  002356                        BGE     2$              ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
```

```
 6018                                   ;+
 6019                                   ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
 6020                                   ;   ACTIVE LINES.  BEFORE WRITING EACH PATTERN, CLEAR ALL THE BITS.
 6021                                   ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1603 - 1605 <<<<<.
 6022                                   ;-
 6023 031472  005267  152266                    INC     ERRNBR          ;SET THE ERROR NUMBER TO 1603.
 6024 031476  005003                            CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
 6025 031500  012704  000002                    MOV     #2,R4           ;INDICATE R/W ACCESS, CLEAR FIRST.
 6026 031504  004767  167322                   JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6027                                   ;+
 6028                                   ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
 6029                                   ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
 6030                                   ;-
 6031 031510  005767  150564                    TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
 6032 031514  001017                           BNE     60$             ;EXIT IF IT IS.
 6033                                   ;+
 6034                                   ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
 6035                                   ;   ACTIVE LINES.  BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
 6036                                   ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1606 - 1608 <<<<<.
 6037                                   ;-
 6038 031516  012767  003106  152240            MOV     #1606.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
 6039 031524  005003                            CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
 6040 031526  005404                            NEG     R4              ;INDICATE R/W ACCESS, SET FIRST.
 6041 031530  004767  167276                   JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6042                                   ;+
 6043                                   ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
 6044                                   ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
 6045                                   ;-
 6046 031534  005767  150540                    TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
 6047 031540  001005                           BNE     60$             ;EXIT IF IT IS.
 6048                                   ;+
 6049                                   ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
 6050                                   ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1609 <<<<<
 6051                                   ;-
 6052 031542  012767  003111  152214            MOV     #1609.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
 6053 031550  004767  167510                   JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
 6054 031554  005067  150522          60$:      CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 6055 031560                                    ENDTST
      031560                                                                    L10044:
      031560  104401                                                                TRAP    C$ETST
```

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

                                        BGNTST
                                                                                        T16::
                                         SETPRI #PRIO5          ;ALLOW THE LTC TO INTERRUPT.
                                                                                        MOV     #PRIO5,R0
                                                                                        TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
                                         MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (17)
                                         MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
                                         MOV    #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
                                         MOV    #1701.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
                                         MOV    #EM1701,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
                                         CLR    ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
                                         MOV    #ERCNTB,R0
                                        JSR     PC,CLR16W       ;CLEAR THE ERROR COUNTER TABLE.
                                         CLR    EXOERR          ;CLEAR THE "EXIT ON ERROR" FLAG
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1701 <<<<<.
                                ;-
                                        JSR     PC,RESETT       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
                                        BCS     .+6             ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
                                        JMP     60$             ;YES,  EXIT THE TEST.
                                ;+
                                ; THE READ/MODIFY/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR IS
                                ;  NOT TESTED THIS THIS FORM OF ACCESS IS ILLEGAL.
                                ;-


                                ;+
                                ; 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 <<<<<.
                                ;-
                                         MOV    #1703.,ERRNBR   ;SET THE ERROR NUMBER TO 1703.
                                         CLR    R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
                                         MOV    #1,R4           ;INDICATE R/M/W ACCESS, CLEAR FIRST.
                                        JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                ;+
                                ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
                                ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
                                ;-
                                         TST    EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
                                        BNE     60$             ;EXIT IF IT IS.
                                ;+
                                ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
                                ;   ACTIVE LINES USING R/M/W.  BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
                                ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1706 - 1708 <<<<<.
```

```
6111                                            ;-
6112 031706  012767  003252  152050             MOV     #1706.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6113 031714  005003                             CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6114 031716  005404                             NEG     R4              ;INDICATE R/M/W ACCESS, SET FIRST.
6115 031720  004767  167106                     JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6116                                            ;+
6117                                            ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6118                                            ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
6119                                            ;-
6120 031724  005767  150350                     TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
6121 031730  001005                             BNE     60$             ;EXIT IF IT IS.
6122
6123                                            ;+
6124                                            ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6125                                            ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1709 <<<<<
6126                                            ;-
6127 031732  012767  003255  152024             MOV     #1709.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
6128 031740  004767  167320                     JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
6129 031744  005067  150332             60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
6130 031750                                     ENDTST
     031750                                                                     L10045:
     031750  104401                                                                TRAP    C$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.
 031752  012700  000240                                                                 MOV     #PRI05,R0
 031756  104441                                                                        TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 031760  012767  000021  150332         MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (18)
 031766  012767  177777  150306         MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
 031774  012767  000001  151760         MOV     #1,ERRTYP      ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
 032002  012767  003411  151754         MOV     #1801.,ERRNBR  ;SET UP ERROR NUMBER IN THE ERROR TABLE.
 032010  012767  012413  151750         MOV     #EM1801,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
 032016  005067  150434                 CLR     ERSMRF         ;CLEAR THE ERROR SUMMARY FLAGS.
 032022  012700  002460                 MOV     #ERCNTB,R0
 032026  004767  165726                 JSR     PC,CLR16W      ;CLEAR THE ERROR COUNTER TABLE.
 032032  005067  150242                 CLR     EXOERR         ;CLEAR THE "EXIT ON ERROR" FLAG.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1801 <<<<<.
                                ;-
 032036  004767  167250                 JSR     PC,RESETT      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
 032042  103402                 BCS     .+6            ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
 032044  000167  000212         JMP     60$            ;YES,  EXIT THE TEST.
 032050  012767  003412  151706         MOV     #1802.,ERRNBR  ;SET THE ERROR REPORT NUMBER TO 1802.
                                ;+
                                ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR.
                                ; USE BYTE ACCESSES.
                                ;-
 032056  012702  000017                 MOV     #17,R2         ;SET LOOP COUNT.
 032062  016704  150150                 MOV     CSRA,R4        ;GET CSR ADDRESS.
 032066  110214                 2$:     MOVB    R2,(R4)        ;WRITE COUNT TO CSR.
 032070  111401                 MOVB    (R4),R1        ;READ BACK THE CONTENTS OF THE CSR
 032072  042701  177760         BIC     #177760,R1     ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
 032076  020102                 CMP     R1,R2          ;CHECK FOR CORRECT DATA WRITTEN/READ.
                                BEQ     4$             ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
 032102  012767  016352  151660         MOV     #ER1601,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
 032110  005003                 CLR     R3             ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
 032112  005005                 CLR     R5             ;CAUSE REPORT OF LINE 0.
 032114                         ERROR                  ;  >>>>> ERROR # 1802 <<<<<
 032114  104460                                                                        TRAP    C$ERROR
                                ;+
                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                ;-
 032116  032767  000100  150074         BIT     #BIT06,OPTION  ;EXIT WITH TEST FAILURE MESSAGE IF
 032124  001456                 BEQ     60$            ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                        ;DURING THE SOFTWARE QUESTIONS.
```

```
6185 032126  005302                    4$:     DEC     R2              ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
6186 032130  002356                            BGE     2$              ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
6187                                    ;+
6188                                    ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6189                                    ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
6190                                    ;   EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6191                                    ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1803 - 1805 <<<<<.
6192                                    ;-
6193 032132  005267  151626                      INC     ERRNBR          ;SET THE ERROR NUMBER TO 1803.
6194 032136  012703  177777                      MOV     #-1,R3          ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6195 032142  012704  000002                      MOV     #2,R4           ;INDICATE R/W ACCESS, CLEAR FIRST.
6196 032146  004767  166660                      JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6197                                    ;+
6198                                    ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6199                                    ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
6200                                    ;-
6201 032152  005767  150122                      TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
6202 032156  001041                            BNE     60$             ;EXIT IF IT IS.
6203
6204                                    ;+
6205                                    ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6206                                    ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
6207                                    ;   EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6208                                    ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1806 - 1808 <<<<<.
6209                                    ;-
6210 032160  012767  003416  151576          MOV     #1806.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6211 032166  005403                            NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6212 032170  004767  166636                      JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6213                                    ;+
6214                                    ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6215                                    ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
6216                                    ;-
6217 032174  005767  150100                      TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
6218 032200  001030                            BNE     60$             ;EXIT IF IT IS.
6219                                    ;+
6220                                    ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6221                                    ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
6222                                    ;   EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6223                                    ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1809 - 1811 <<<<<.
6224                                    ;-
6225 032202  012767  003421  151554          MOV     #1809.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6226 032210  005403                            NEG     R3              ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6227 032212  005404                            NEG     R4              ;INDICATE R/W ACCESS, SET FIRST.
6228 032214  004767  166612                      JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6229                                    ;+
6230                                    ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6231                                    ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
6232                                    ;-
6233 032220  005767  150054                      TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
6234 032224  001016                            BNE     60$             ;EXIT IF IT IS.
6235                                    ;+
6236                                    ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6237                                    ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
6238                                    ;   EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6239                                    ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1812 - 1814 <<<<<.
6240                                    ;-
6241 032226  012767  003424  151530          MOV     #1812.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
```

```
6242 032234  005403                        NEG    R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6243 032236  004767  166570                JSR    PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6244                                  ;+
6245                                  ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6246                                  ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
6247                                  ;-
6248 032242  005767  150032                TST    EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
6249 032246  001005                        BNE    60$             ;EXIT IF IT IS.
6250                                  ;+
6251                                  ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6252                                  ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1815 <<<<<
6253                                  ;-
6254 032250  012767  003427  151506          MOV    #1815.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
6255 032256  004767  167002                JSR    PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
6256 032262  005067  150014          60$:   CLR    CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
6257 032266                                ENDTST
     032266                                                                  L10046:
     032266  104401                                                                TRAP    C$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.
                                ;*
                                ;-- ****************************************************************************

032270                                  BGNTST
032270                                                                  T18::
032270                           SETPRI #PRI05          ;ALLOW THE LTC TO INTERRUPT.
032270  012700  000240                                                  MOV     #PRI05,R0
032274  104441                                                          TRAP    C$SPRI
        000022                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
032276  012767  000022  150014    MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (19)
032304  012767  177777  147770    MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
032312  012767  000001  151442    MOV     #1,ERRTYP      ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
032320  012767  003555  151436    MOV     #1901.,ERRNBR  ;SET UP ERROR NUMBER IN THE ERROR TABLE.
032326  012767  012467  151432    MOV     #EM1901,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
032334  005067  150116            CLR     ERSMRF         ;CLEAR THE ERROR SUMMARY FLAGS.
032340  012700  002460            MOV     #ERCNTB,R0
032344  004767  165410            JSR     PC,CLR16W      ;CLEAR THE ERROR COUNTER TABLE.
032350  005067  147724            CLR     EXOERR         ;CLEAR THE "EXIT ON ERROR" FLAG.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1901 <<<<<.
                                ;-
032354  004767  166732                    JSR     PC,RESETT      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
032360  103402                            BCS     .+6            ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
032362  000167  000136                    JMP     60$            ;YES,  EXIT THE TEST.
032366  012767  003557  151370            MOV     #1903.,ERRNBR  ;SET THE ERROR REPORT NUMBER TO 1903.
                                ;+
                                ; THE READ/MODIFY/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR IS NOT
                                ; TESTED SINCE THIS IS AN ILLEGAL FORM OF ACCESS TO THIS REGISTER.
                                ;-


                                ;+
                                ; 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 <<<<<.
                                ;-
032374  005267  151364                    INC     ERRNBR         ;SET THE ERROR NUMBER TO 1903.
032400  012703  177777                    MOV     #-1,R3         ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
032404  012704  000001                    MOV     #1,R4          ;INDICATE R/M/W ACCESS, CLEAR FIRST.
032410  004767  166416                    JSR     PC,REGTST      ;WRITE AND VERIFY DATA PATTERNS.
                                ;+
                                ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
                                ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
                                ;-
032414  005767  147660                    TST     EXOERR         ;IS THE "EXIT ON ERROR" FLAG SET ?
032420  001041                            BNE     60$            ;EXIT IF IT IS.
                                ;+
                                ; 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 <<<<<.
                                ;-
032422  012767  003562  151334          MOV     #1906.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
032430  005403                          NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
032432  004767  166374                  JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                ;+
                                ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
                                ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
                                ;-
032436  005767  147636                  TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
032442  001030                          BNE     60$             ;EXIT IF IT IS.
                                ;+
                                ; 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 <<<<<.
                                ;-
032444  012767  003565  151312          MOV     #1909.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
032452  005403                          NEG     R3              ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
032454  005404                          NEG     R4              ;INDICATE R/M/W ACCESS, SET FIRST.
032456  004767  166350                  JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                ;+
                                ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
                                ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
                                ;-
032462  005767  147612                  TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
032466  001016                          BNE     60$             ;EXIT IF IT IS.
                                ;+
                                ; 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 <<<<<.
                                ;-
032470  012767  003570  151266          MOV     #1912.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
032476  005403                          NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
032500  004767  166326                  JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
                                ;+
                                ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
                                ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
                                ;-
032504  005767  147570                  TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
032510  001005                          BNE     60$             ;EXIT IF IT IS.
                                ;+
                                ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
                                ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1915 <<<<<
                                ;-
032512  012767  003573  151244          MOV     #1915.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
032520  004767  166540                  JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
032524  005067  147552          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
032530                                  ENDTST
032530                                                                  L10047:
032530  104401                                                                  TRAP    C$ETST
```

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

                                        BGNTST
                                                                                        T19::
                                         SETPRI #PRI05          ;ALLOW THE LTC TO INTERRUPT.
032532  012700  000240                                                                  MOV     #PRI05,R0
032536  104441                                                                         TRAP    C$SPRI
        000023                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
032540  012767  000023  147552         MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (20)
032546  012767  177777  147526         MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
032554  012767  000001  151200         MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
032562  012767  003721  151174         MOV     #2001.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
032570  012767  012552  151170         MOV     #EM2001,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 2001 <<<<<.
                                ;-
032576  004767  166510                 JSR     PC,RESETT       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
032602  103016                         BCC     60$             ;FATAL RESET ERROR?  YES,  EXIT THE TEST.
                                ;+
                                ; READ THE STAT REGISTER ID BIT AND VERIFY THAT IT IS CLEAR.
                                ;-
032604  017701  147434                  MOV     @FSLSA,R1       ;READ THE STAT REGISTER CONTENTS.
032610  032701  000400                  BIT     #BIT8,R1        ;CHECK THE ID BIT.
032614  001011                         BNE     60$             ;ID BIT SET?  YES, EXIT THE TEST.
032616  012767  003722  151140          MOV     #2002.,ERRNBR   ;NO,  SET THE ERROR REPORT NUMBER TO 2002.
032624  012701  012622                  MOV     #EM2002,R1      ;GET THE PROPER ERROR MESSAGE.
032630  012767  016072  151132          MOV     #ER0503,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
032636                                 ERROR                   ;ERROR NUMBER >>>>> 2002 <<<<<
032636  104460                                                                         TRAP    C$ERROR
032640  005067  147436          60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
032644                                 ENDTST
032644                                                                          L10050:
032644  104401                                                                         TRAP    C$ETST
```

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

032646                                  BGNTST
032646                                                                          T20::
032646                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
032646  012700  000240                                                        MOV     #PRI05,R0
032652  104441                                                                TRAP    C$SPRI
000024                                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
032654  012767  000024  147436           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (23)
032662  012767  177777  147412           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
032670  012767  000001  151064           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
032676  012767  004375  151060           MOV     #2301.,ERRNBR  ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
032704  012767  012675  151054           MOV     #EM2301,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
032712  012767  017156  151050           MOV     #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 2301 <<<<<.
                                ;-
032720  004767  165012                  JSR     PC,CLNRST      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
032724  103114                          BCC     60$            ;RESET FAILURE?, ABORT THIS TEST.
                                ;*
                                ; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
                                ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
                                ; 2 STOP BITS.
                                ; ENABLE TRANSMITTERS ON ALL LINES.
                                ;-
032726  016705  147272                   MOV     ACTLNS,R5      ;PASS THE ACTIVE LINE BIT MAP.
032732  012700  000200                   MOV     #200,R0        ;PASS THE LNCTRL CONTENTS.
032736  004767  170470                  JSR     PC,WTWLNC      ;INITIALISE THE LNCTRL REGISTERS.
032742  012700  177670                   MOV     #177670,R0     ;PASS THE LPR CONTENTS.
032746  004767  170510                  JSR     PC,WTWLPR      ;INITIALSE THE LPR REGISTERS ON ALL LINES.
032752  012704  000012                   MOV     #10.,R4        ;PASS DELAY TIME OF 10 MILLI-SECONDS.
032756  004767  165076                  JSR     PC,DELAY       ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
032762  012705  177777                   MOV     #MAPLNS,R5     ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
032766  004767  167642                  JSR     PC,TXENBL      ;ENABLE TRANSMITTERS ON ALL LINES.
                                ;*
                                ; TEST ALL ACTIVE LINES INDIVIDUALLY.
                                ; DISABLE TRANSMISSION ON EACH ACTIVE LINE.
                                ;-
032772  012703  000001                   MOV     #1,R3          ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
032776  005004                           CLR     R4             ;CLEAR THE LINE NUMBER COUNTER.
033000  012767  004376  150756  2$:      MOV     #2302.,ERRNBR  ;SET THE ERROR NUMBER TO 2302.
033006  030367  147212                   BIT     R3,ACTLNS      ;CHECK IF THE LINE IS ACTIVE.
033012  001453                          BEQ     6$             ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
                                ;*
                                ; CLEAR THE TX_ENABLE BIT IN TBUFFAD2 REGISTER.
                                ; SELECT THE LINE UNDER TEST.
                                ;   VERIFY IT IS CLEAR, REPORT ERROR IF SET.
                                ;-
```

```
 6456 033014  010305                          MOV    R3,R5           ;PASS THE BIT MAP OF THE LINE UNDER TEST.
 6457 033016  004767  167516                  JSR    PC,TXDSBL       ;DISABLE TRANSMISSION ON THE LINE UNDER TEST.
 6458 033022  010477  147210                  MOV    R4,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
 6459 033026  005777  147220                  TST    @TXAD2A         ;VERIFY THE TX_ENABLE BIT IS SET.
 6460 033032  100433                          BMI    4$              ;GO REPORT ERROR IF TX_ENABLE BIT SET.
 6461                                         ;+
 6462                                         ; WRITE DATA BYTE (ASCII <LF>) TO THE OUTPUT FIFO.
 6463                                         ; WAIT FOR A TX_ACTION TO BE RETURNED, REPORT ERROR IF A TX_ACTION
 6464                                         ; IS FOUND BEFORE TIME-OUT OCCURS.
 6465                                         ;-
 6466 033034  012767  004377  150722          MOV    #2303.,ERRNBR   ;SET ERROR NUMBER TO 2303.
 6467 033042  112777  000012  147174          MOVB   #12,@FDATA      ;WRITE THE DATA BYTE TO THE DUT'S OUTPUT FIFO.
 6468 033050  012701  170003                  MOV    #170003,R1      ;TEST BIT 15, TIMEOUT OF 3 MILLI SECS.
 6469 033054  016702  147156                  MOV    CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 6470 033060  004767  170064                  JSR    PC,WAIBIS       ;WAIT FOR TX_ACTION TO COME BACK.
 6471 033064  103416                          BCS    4$              ;GO REPORT ERROR IF A TX-ACTION FOUND.
 6472                                         ;+
 6473                                         ; WAIT FOR THE DATA TO APPEAR IN THE FIFO, REPORT ERROR IF DATA FOUND.
 6474                                         ;-
 6475 033066  005267  150672                  INC    ERRNBR          ;SET ERROR NUMBER TO 2304.
 6476 033072  012701  070012                  MOV    #70012,R1       ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
 6477 033076  016702  147134                  MOV    CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 6478 033102  004767  170042                  JSR    PC,WAIBIS       ;WAIT FOR RX_DATA_AVAILABLE TO SET.
 6479 033106  103405                          BCS    4$              ;REPORT ERROR IF DATA RECEIVED IN THE FIFO.
 6480 033110  005267  150650                  INC    ERRNBR          ;SET ERROR NUMBER TO 2305.
 6481 033114  017702  147120                  MOV    @RBUFA,R2       ;READ THE DATA FROM THE FIFO.
 6482 033120  100010                          BPL    6$              ;SKIP ERROR REPORT IF DATA ISN'T THERE.
 6483
 6484 033122  010401                  4$:     MOV    R4,R1           ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
 6485 033124  012702  012742                  MOV    #EM2302,R2      ;PASS THE MESSAGE TO BE REPORTED.
 6486                                                                 ;"TX_ENABLE BIT BAD ON LINE: NN".
 6487 033130                                  ERROR                  ;                >>>>> ERROR <<<<<.
      033130  104460                                                                      TRAP    C$ERROR
 6488
 6489 033132  032767  000100  147060          BIT    #BIT06,OPTION   ;EXIT THE TEST, WITH THE TEST FAILED MESSAGE.
 6490 033140  001406                          BEQ    60$             ;IF EXTENDED ERROR REPORTING HAS NOT BEEN
 6491                                                                 ;REQUESTED.
 6492
 6493                                 ;+
 6494                                 ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
 6495                                 ;-
 6496 033142  000241                  6$:     CLC                    ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
 6497 033144  006103                          ROL    R3              ;SHIFT THE BIT MAP FOR THE NEXT LINE.
 6498 033146  005204                          INC    R4              ;INCREMENT THE LINE NUMBER COUNTER.
 6499 033150  020427  000020                  CMP    R4,#NUMLNS      ;HAVE ALL THE LINES BEEN TESTED?.
 6500 033154  002711                          BLT    2$              ;NO; BRANCH TO TEST THE NEXT LINE.
 6501
 6502 033156  005067  147120          60$:    CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 6503 033162                                  ENDTST
      033162                                                                       L10051:
      033162  104401                                                                      TRAP    C$ETST
```

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

033164                                  BGNTST
033164                                                                  T21::
033164                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
033164  012700  000240                                                  MOV     #PRI05,R0
033170  104441                                                         TRAP    C$SPRI
        000025                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
033172  012767  000025  147120           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (24)
033200  012767  177777  147074           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
033206  012767  000001  150546           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
033214  012767  004541  150542           MOV     #2401.,ERRNBR  ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
033222  012767  013000  150536           MOV     #EM2401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
033230  012767  017156  150532           MOV     #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 2401 <<<<<.
                                ;-
033236  004767  164474                   JSR     PC,CLNRST      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
033242  103133                          BCC     60$            ;RESET FAILURE?, ABORT THIS TEST.
                                ;*
                                ; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
                                ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
                                ; 2 STOP BITS.
                                ; DISABLE TRANSMITTERS ON ALL LINES.
                                ;-
033244  016705  146754                   MOV     ACTLNS,R5      ;PASS THE ACTIVE LINE BIT MAP.
033250  012700  000200                   MOV     #200,R0        ;PASS THE LNCTRL CONTENTS.
033254  004767  170152                   JSR     PC,WTWLNC      ;INITIALISE THE LNCTRL REGISTERS.
033260  012700  177670                   MOV     #177670,R0     ;PASS THE LPR CONTENTS.
033264  004767  170172                   JSR     PC,WTWLPR      ;INITIALSE THE LPR REGISTERS ON ALL LINES.
033270  012704  000012                   MOV     #10.,R4        ;PASS DELAY TIME OF 10 MILLI-SECONDS.
033274  004767  164560                   JSR     PC,DELAY       ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
033300  012705  177777                   MOV     #MAPLNS,R5     ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
033304  004767  167230                   JSR     PC,TXDSBL      ;DISABLE TRANSMITTERS ON ALL LINES.
                                ;*
                                ; TEST ALL ACTIVE LINES INDIVIDUALLY.
                                ; ENABLE TRANSMISSION ON EACH ACTIVE LINE.
                                ;-
033310  012703  000001                   MOV     #1,R3          ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
033314  005004                           CLR     R4             ;CLEAR THE LINE NUMBER COUNTER.
033316  012767  004542  150440  2$:      MOV     #2402.,ERRNBR  ;SET THE ERROR NUMBER TO 2402.
033324  030367  146674                   BIT     R3,ACTLNS      ;CHECK IF THE LINE IS ACTIVE.
033330  001467                          BEQ     8$             ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
                                ;*
                                ; SELECT THE LINE UNDER TEST.
                                ;  SET THE TX_ENABLE BIT IN TBUFFAD2 REGISTER.
                                ;   VERIFY IT IS SET, REPORT ERROR IF CLEAR.
                                ;-
```

```
 6559 033332  010305                          MOV     R3,R5           ;PASS THE BIT MAP OF THE LINE UNDER TEST.
 6560 033334  004767  167274                  JSR     PC,TXENBL       ;ENABLE TRANSMISSION ON THE LINE UNDER TEST.
 6561 033340  012705  000012                  MOV     #10.,R5         ;SET TXCHAR/LOOP COUNT TO 10.
 6562 033344  010477  146666                  MOV     R4,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
 6563 033350  005777  146676                  TST     @TXAD2A         ;VERIFY THE TX_ENABLE BIT IS SET.
 6564 033354  100045                          BPL     6$              ;GO REPORT ERROR IF TX_ENABLE BIT CLEAR.
 6565                                        ;+
 6566                                        ; WRITE DATA BYTE (ASCII <LF>) TO OUTPUT FIFO.
 6567                                        ; WAIT FOR A TX_ACTION TO BE RETURNED, REPORT ERROR IF NO TX_ACTION
 6568                                        ; FOUND BEFORE TIME-OUT OCCURS.
 6569                                        ;-
 6570 033356  012767  004543  150400  4$:     MOV     #2403.,ERRNBR   ;SET ERROR NUMBER TO 2403.
 6571 033364  112777  000012  146652          MOVB    #12,@FDATA      ;WRITE THE DATA BYTE TO THE DUT'S OUTPUT FIFO.
 6572 033372  012701  170004                  MOV     #170004,R1      ;TEST BIT 15, TIMEOUT OF 4 MILLI SECS.
 6573 033376  016702  146634                  MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 6574 033402  004767  167542                  JSR     PC,WAIBIS       ;WAIT FOR TX_ACTION TO COME BACK.
 6575 033406  103030                          BCC     6$              ;GO REPORT ERROR IF NO TX-ACTION FOUND.
 6576                                        ;+
 6577                                        ; WAIT FOR THE DATA TO APPEAR IN THE FIFO, REPORT ERROR IF TIME-OUT.
 6578                                        ;-
 6579 033410  005267  150350                  INC     ERRNBR          ;SET ERROR NUMBER TO 2404.
 6580 033414  012701  070012                  MOV     #70012,R1       ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
 6581 033420  016702  146612                  MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 6582 033424  004767  167520                  JSR     PC,WAIBIS       ;WAIT FOR RX_DATA_AVAILABLE TO SET.
 6583 033430  103017                          BCC     6$              ;REPORT ERROR IF NO DATA RECEIVED IN THE FIFO.
 6584 033432  005267  150326                  INC     ERRNBR          ;SET ERROR NUMBER TO 2405.
 6585 033436  017702  146576                  MOV     @RBUFA,R2       ;READ THE DATA FROM THE FIFO.
 6586 033442  100012                          BPL     6$              ;GO REPORT ERROR IF THER IS'NT ANY DATA THERE.
 6587 033444  005267  150314                  INC     ERRNBR          ;SET ERROR NUMBER TO 2406.
 6588 033450  000302                          SWAB    R2              ;PUT THE LINE NUMBER IN THE LOW BYTE.
 6589 033452  042702  177760                  BIC     #177760,R2      ;CLEAR THE UNWANTED BITS.
 6590 033456  020204                          CMP     R2,R4           ;DID THE DATA COME FROM THE CORRECT LINE?.
 6591 033460  001003                          BNE     6$              ;NO; GO REPORT THE ERROR.
 6592 033462  005305                          DEC     R5              ;DECREMENT THE TXCHAR/LOOP COUNTER.
 6593 033464  001334                          BNE     4$              ;LOOP TO TX THE NEXT CHAR.
 6594 033466  000410                          BR      8$              ;GO TEST THE NEXT LINE.
 6595
 6596 033470  010401                  6$:     MOV     R4,R1           ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
 6597 033472  012702  012742                  MOV     #EM2302,R2      ;PASS THE MESSAGE TO BE REPORTED.
 6598                                                                 ;"TX_ENABLE BIT BAD ON LINE: NN".
 6599 033476                                  ERROR                   ;                >>>>> ERROR <<<<<.
      033476  104460                                                                      TRAP    C$ERROR
 6600
 6601 033500  032767  000100  146512          BIT     #BIT06,OPTION   ;EXIT THE TEST IF EXTENDED ERROR REPORTING
 6602 033506  001411                          BEQ     60$             ;HAS NOT BEEN ENABLED, SINCE THE TEST HAS FAILED.
 6603
 6604                                        ;+
 6605                                        ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
 6606                                        ;-
 6607 033510  010305                  8$:     MOV     R3,R5           ;PASS THE BIT MAP OF THE LINE UNDER TEST.
 6608 033512  004767  167022                  JSR     PC,TXDSBL       ;CLEAR THE TX_ENABLE BIT ON THIS LINE.
 6609 033516  000241                          CLC                     ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
 6610 033520  006103                          ROL     R3              ;SHIFT THE BIT MAP FOR THE NEXT LINE.
 6611 033522  005204                          INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
 6612 033524  020427  000020                  CMP     R4,#NUMLNS      ;HAVE ALL THE LINES BEEN TESTED?.
 6613 033530  002672                          BLT     2$              ;NO; BRANCH TO TEST THE NEXT LINE.
 6614
```

```
6615 033532  005067  146544          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6616 033536                                  ENDTST
     033536                                                          L10052:
     033536  104401                                                          TRAP    C$ETST
```

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

033540                                  BGNTST
033540                                                                          T22::
033540                                  SETPRI  #PRI05          ;ALLOW THE LTC TO INTERRUPT.
033540  012700  000240                                                        MOV     #PRI05,R0
033544  104441                                                                TRAP    C$SPRI
        000026                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
033546  012767  000026  146544           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (26)
033554  012767  177777  146520           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
033562  012767  000001  150172           MOV     #1.,ERRTYP     ;SET ERROR FATAL ERROR TYPE IN ERROR TABLE.
033570  012767  003101  150166           MOV     #1601.,ERRNBR  ;SET FIRST ERROR REPORT NUMBER IN ERROR TABLE.
033576  012767  013043  150162           MOV     #EM2601.,ERRMSG ;SET TEST ERROR MESSAGE IN ERROR TABLE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS FROM >>>>> 2601 THRU 2602 <<<<<.
                                ;-
033604  004767  165502                  JSR     PC,RESETT      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
033610  103402                          BCS     2$             ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
033612  000167  001044                  JMP     60$            ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
033616  012767  005053  150140  2$:      MOV     #2603.,ERRNBR  ;SET THE ERROR REPORT NUMBER TO 2603.
                                ;+
                                ; ENABLE TRANSMITTERS ON ALL LINES.
                                ;-
033624  012705  177777          4$:      MOV     #MAPLNS,R5     ;PASS ACTIVE LINE BIT MAP.
033630  004767  167000                  JSR     PC,TXENBL      ;ENABLE TRANSMISSION ON ALL LINES.
                                ;+
                                ; TEST RECEPTION INTERRUPTS.
                                ;    SET UP FOR RX AND TX INTERRUPTS:
                                ;    RX INTERRUPT SERVICE ROUTINE INPUTS A CHAR AND COUNTS THE INTERRUPT.
                                ;    TX INTERRUPT SERVICE ROUTINE COUNTS TX INTERRUPTS.
                                ;-
033634  005067  146450                   CLR     RXINTC         ;CLEAR THE RX INTERRUPT COUNTER.
033640  005067  146446                   CLR     RXINTF         ;CLEAR THE RX INTERRUPT FLAGS.
033644  005067  146452                   CLR     TXINTC         ;CLEAR THE TX INTERRUPT COUNTER.
033650  012767  002722  146420           MOV     #BUFBAS,BUFPTR ;LOAD THE BUFFER PTR WITH THE BUFFER BASE ADR.
033656                                  SETVEC  RXVECA,#RXINPT,#PRI06 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
033656  012746  000300                                                        MOV     #PRI06,-(SP)
033662  012746  023716                                                        MOV     #RXINPT,-(SP)
033666  016746  146334                                                        MOV     RXVECA,-(SP)
033672  012746  000003                                                        MOV     #3,-(SP)
033676  104437                                                                TRAP    C$SVEC
033700  062706  000010                                                        ADD     #10,SP
033704                                  SETVEC  TXVECA,#CACHTX,#PRI06 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
033704  012746  000300                                                        MOV     #PRI06,-(SP)
033710  012746  023540                                                        MOV     #CACHTX,-(SP)
```

```
        033714  016746  146310                                                          MOV     TXVECA,-(SP)
        033720  012746  000003                                                          MOV     #3,-(SP)
        033724  104437                                                                  TRAP    C$SVEC
        033726  062706  000010                                                          ADD     #10,SP
  6664  033732                          SETPRI  #PRI04          ;ALLOW DEVICE INTERRUPTS.
        033732  012700  000200                                                          MOV     #PRI04,R0
        033736  104441                                                                  TRAP    C$SPRI
  6665                                ;+
  6666                                ;ENABLE RECEPTION INTERRUPTS.
  6667                                ;DELAY 4 MS TO ALLOW TIME FOR THE INTERRUPTS TO TAKE PLACE.
  6668                                ;DISABLE RECEPTION INTERRUPTS.
  6669                                ;-
  6670  033740  004767  166242           JSR     PC,RXIE1        ;ENABLE THE RECEPTION INTERRUPTS.
  6671  033744  012704  000004            MOV     #4,R4           ;PASS 4 MS COUNT TO THE DELAY ROUTINE.
  6672  033750  004767  164104           JSR     PC,DELAY        ;DELAY 4 MILLI-SECONDS.
  6673  033754  004767  166166           JSR     PC,RXIE0        ;DISABLE RECEPTION INTERRUPTS.
  6674                                ;+
  6675                                ; VERIFY THAT THE CORRECT INTERRUPTS TOOK PLACE.
  6676                                ;    TEST THE INT COUNTER TO VERIFY THAT INTERRUPTS TOOK PLACE.
  6677                                ;-
  6678  033760  005767  146324           TST     RXINTC          ;CHECK THE RX INTERRUPT COUNT.
  6679  033764  001017                   BNE     6$              ;SKIP THE FOLLOWING ERRORS IF COUNT <> 0.
  6680                                ;+
  6681                                ; DETERMINE REASON FOR NO RX INTERRUPTS AND PRINT PROPER ERROR MESSAGE.
  6682                                ;-
  6683  033766  012701  013260           MOV     #EM2604,R1      ;SET UP MSG IN CASE "RX.DATA.AVAIL IS CLR".
  6684  033772  032777  000200  146236   BIT     #BIT7,@CSRA     ;TEST THE RX.DATA.AVAIL BIT OF THE CSR.
  6685  034000  001416                   BEQ     8$              ;GO REPORT ERROR IF RX.DATA.AVAIL IS CLR.
  6686  034002  012701  013172           MOV     #EM2603,R1      ;SET UP MSG IN CASE "DATA.VALID IS CLEAR".
  6687  034006  032777  100000  146224   BIT     #BIT15,@RBUFA   ;TEST THE DATA.VALID BIT OF THE FIFO.
  6688  034014  001410                   BEQ     8$              ;GO REPORT ERROR IF DATA.VALID IS CLEAR.
  6689  034016  012701  013101           MOV     #EM2602,R1      ;SET UP MSG,"DATA.VALID IS SET".
  6690  034022  000405                   BR      8$              ;GO REPORT THE ERROR.
  6691                                ;+
  6692                                ; IF RX INTS OCCURRED WITH RX.DATA.AVAIL CLEAR, REPORT THE ERROR.
  6693                                ;-
  6694  034024  005767  146262     6$:    TST     RXINTF          ;CHECK THE RX INTERRUPT FLAGS.
  6695  034030  100014                   BPL     10$             ;SKIP THE ERROR IF FLAG IS CLEAR.
  6696  034032  012701  013354           MOV     #EM2605,R1      ;SET UP THE PROPER MESSAGE.
  6697                                ;+
  6698                                ; REPORT THE ERROR WHICH HAS BEEN FOUND.
  6699                                ;-
  6700  034036                     8$:    ERRDF   2603,EM2601,ER0503;             >>>>> ERROR #2603 <<<<<.
        034036  104455                                                                  TRAP    C$ERDF
        034040  005053                                                                  .WORD   2603
        034042  013043                                                                  .WORD   EM2601
        034044  016072                                                                  .WORD   ER0503
  6701
  6702                                ;+
  6703                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
  6704                                ;-
  6705
  6706  034046  032767  000100  146144     BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
  6707  034054  001002                     BNE     .+6             ;
  6708  034056  000167  000556             JMP     34$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
  6709
  6710                                ;+
```

```
 6711                                   ; VERIFY THAT NO TX INTERRUPTS HAVE BEEN GENERATED SO FAR IN THIS TEST.
 6712                                   ;-
 6713 034062  016702  146234            10$:    MOV     TXINTC,R2       ;LOAD # OF TX INTERRUPTS FOR ER0504 RTN.
 6714 034066  001414                            BEQ     12$             ;SKIP ERROR IF NO TX INTERRUPTS.
 6715                                       ;REPORT "TX INTERRUPTS(S) RECEIVED WITH TX INTERRUPTS DISABLED."
 6716 034070  012701  007521                    MOV     #EM0526,R1      ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
 6717 034074                                    ERRDF   2604,EM2601,ER0504;             >>>>> ERROR #2604 <<<<<.
      034074  104455                                                                            TRAP    C$ERDF
      034076  005054                                                                            .WORD   2604
      034100  013043                                                                            .WORD   EM2601
      034102  016130                                                                            .WORD   ER0504
 6718
 6719                                       ;+
 6720                                       ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
 6721                                       ;-
 6722
 6723 034104  032767  000100  146106            BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
 6724 034112  001002                            BNE     .+6             ;
 6725 034114  000167  000520                    JMP     34$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
 6726
 6727                                   ;+
 6728                                   ; CLEAN OUT THE INTERRUPT VECTORS USED IN THIS TEST.
 6729                                   ;-
 6730 034120                            12$:    SETPRI  #PRI06          ;DISABLE DEVICE INTERRUPTS.
      034120  012700  000300                                                                    MOV     #PRI06,R0
      034124  104441                                                                            TRAP    C$SPRI
 6731 034126                                    CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
      034126  016700  146074                                                                    MOV     RXVECA,R0
      034132  104436                                                                            TRAP    C$CVEC
 6732 034134                                    CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
      034134  016700  146070                                                                    MOV     TXVECA,R0
      034140  104436                                                                            TRAP    C$CVEC
 6733                                   ;+
 6734                                   ; TEST TRANSMISSION INTERRUPTS.
 6735                                   ;    SET UP FOR RX AND TX INTERRUPTS:
 6736                                   ;    RX INTERRUPT SERVICE ROUTINE COUNTS RX INTERRUPTS.
 6737                                   ;    TX INTERRUPT SERVICE ROUTINE COUNTS THE INTERRUPT AND SETS FLAGS.
 6738                                   ;-
 6739 034142  005067  146142                    CLR     RXINTC          ;CLEAR THE RX INTERRUPT COUNTER.
 6740 034146  005067  146150                    CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNTER.
 6741 034152  005067  146146                    CLR     TXINTF          ;CLEAR THE RX INTERRUPT FLAGS.
 6742 034156                                    SETVEC  RXVECA,#CACHRX,#PRI06 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
      034156  012746  000300                                                                    MOV     #PRI06,-(SP)
      034162  012746  023512                                                                    MOV     #CACHRX,-(SP)
      034166  016746  146034                                                                    MOV     RXVECA,-(SP)
      034172  012746  000003                                                                    MOV     #3,-(SP)
      034176  104437                                                                            TRAP    C$SVEC
      034200  062706  000010                                                                    ADD     #10,SP
 6743 034204                                    SETVEC  TXVECA,#TXINTR,#PRI06 ;SET UP INT VECTOR TO TX INT ROUTINE.
      034204  012746  000300                                                                    MOV     #PRI06,-(SP)
      034210  012746  024026                                                                    MOV     #TXINTR,-(SP)
      034214  016746  146010                                                                    MOV     TXVECA,-(SP)
      034220  012746  000003                                                                    MOV     #3,-(SP)
      034224  104437                                                                            TRAP    C$SVEC
      034226  062706  000010                                                                    ADD     #10,SP
 6744 034232                                    SETPRI  #PRI04          ;ALLOW DEVICE INTERRUPTS.
      034232  012700  000200                                                                    MOV     #PRI04,R0
```

```
     034236  104441                                                                  TRAP    C$SPRI
6745                                    ;+
6746                                    ; VERIFY THAT THE TX_ACTION BIT IS CLEAR.
6747                                    ;-
6748 034240  012705  000022                 MOV     #18.,R5         ;INITIALIZE THE LOOP COUNTER.
6749 034244  012701  000144                 MOV     #100.,R1        ;SET 100 MS TIME-OUT.
6750 034250  012702  100000                 MOV     #BIT15,R2       ;SELECT TX_ACTION BIT TO TEST.
6751 034254  016704  145756                 MOV     CSRA,R4         ;PASS DUT CSR AS THE WORD TO TEST.
6752 034260  012703  100000           14$:   MOV     #BIT15,R3       ;WAIT FOR TX_ACTION TO BE SET.
6753 034264  004767  163744                 JSR     PC,MSLOOP       ;WAIT UP TO 100 MS FOR TX_ACTION SET.
6754 034270  103026                        BCC     20$             ;IF TIME-OUT, CONSIDER TX_ACTION CLEAR.
6755 034272  005003                         CLR     R3              ;NOW, WAIT FOR TX_ACTION CLEAR.
6756 034274  004767  163734                 JSR     PC,MSLOOP       ;WAIT UP TO 100 MS FOR TX_ACTION CLEAR.
6757 034300  103005                        BCC     16$             ;IF TIME-OUT, REPORT TX_ACTION WON'T CLEAR.
6758 034302  005305                         DEC     R5              ;DECREMENT THE TX_ACTION SET COUNTER.
6759 034304  001365                        BNE     14$             ;LOOP IF NOT TOO MANY TX_ACTIONS FOUND.
6760                                    ;REPORT "TX_ACTION SET REPEATEDLY AFTER RESET, NO DATA SENT."
6761 034306  012701  013476                 MOV     #EM2607,R1      ;SELECT ERROR MESSAGE.
6762 034312  000402                        BR      18$             ;GO TO REPORT THE ERROR.
6763 034314  012701  013572           16$:   MOV     #EM2608,R1      ;SELECT TX_ACTION STUCK SET MSG.
6764 034320                         18$:   ERRDF   2605,EM2606,ER0503;             >>>>> ERROR #2605 <<<<<.
     034320  104455                                                                  TRAP    C$ERDF
     034322  005055                                                                  .WORD   2605
     034324  013437                                                                  .WORD   EM2606
     034326  016072                                                                  .WORD   ER0503
6765
6766                                    ;+
6767                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
6768                                    ;-
6769 034330  032767  000100  145662         BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
6770 034336  001540                        BEQ     34$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
6771 034340  004767  166424                 JSR     PC,TXIE1        ;ENABLE TX INTERRUPTS FOR THE TX_INT TESTING.
6772 034344  000430                        BR      24$             ;GO TO TEST WITH TX_ACTION SET.
6773                                    ;+
6774                                    ; VERIFY THAT NO INTERRUPTS OCCUR WITH TX_ACTION CLEAR.
6775                                    ;-
6776 034346  004767  166416         20$:   JSR     PC,TXIE1        ;ENABLE TX_INTERRUPTS.
6777 034352  012704  000062                 MOV     #50.,R4         ;PASS 50 MS TIME TO THE DELAY ROUTINE.
6778 034356  004767  163476                 JSR     PC,DELAY        ;DELAY 50 MILLI-SECONDS TO ALLOW INTS TO OCCUR.
6779 034362  005767  145734                 TST     TXINTC          ;TEST THE TX INTERRUPT COUNT.
6780 034366  001417                        BEQ     24$             ;SKIP THE ERROR IF NO TX INTERRUPTS.
6781 034370  012701  013476                 MOV     #EM2607,R1      ;SELECT MESSAGE IN CASE TX INT FLAG CLEAR.
6782 034374  005767  145724                 TST     TXINTF          ;TEST THE TX INTERRUPT FLAGS.
6783 034400  100002                        BPL     22$             ;GO REPORT ERROR IF TX FLAG IS CLEAR.
6784 034402  012701  013643                 MOV     #EM2609,R1      ;TX FLAG IS SET, SELECT PROPER ERROR MESSAGE.
6785                                    ;REPORT "TRANSMIT INTERRUPT TEST ERROR:..."
6786 034406                         22$:   ERRDF   2606,EM2606,ER0503;             >>>>> ERROR #2606 <<<<<.
     034406  104455                                                                  TRAP    C$ERDF
     034410  005056                                                                  .WORD   2606
     034412  013437                                                                  .WORD   EM2606
     034414  016072                                                                  .WORD   ER0503
6787
6788                                    ;+
6789                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
6790                                    ;-
6791 034416  032767  000100  145574         BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
6792 034424  001500                        BEQ     32$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
```

```
6793
6794                                    ;+
6795                                    ; PREPARE TX INTERRUPT COUNTER AND FLAGS.
6796                                    ;-
6797 034426  005067  145670        24$:     CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNT.
6798 034432  005067  145666                 CLR     TXINTF          ;CLEAR THE TX INTERRUPT FLAGS.
6799                                    ;+
6800                                    ; SET UP LINE PARAMETERS FOR TRANSMISSION.
6801                                    ;-
6802 034436  012705  177777                 MOV     #MAPLNS,R5      ;PASS ACTIVE LINES BIT MAP.
6803 034442  012700  000200                 MOV     #200,R0         ;PASS INERT STATE, INTERNAL LOOPBACK.
6804 034446  004767  166760                 JSR     PC,WTWLNC       ;DISABLE RECPTION AND DMA, ETC. ON DUT.
6805 034452  012700  156430                 MOV     #156430,R0      ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
6806 034456  004767  167000                 JSR     PC,WTWLPR       ;WRITE TO ALL LPR REGISTERS.
6807                                    ;+
6808                                    ; SEND A NULL CHAR TO EACH LINE.
6809                                    ;-
6810 034462  016701  145616                 MOV     IESTAT,R1       ;SET UP THE STATE OF THE INTERRUPT ENABLE BITS.
6811 034466  005002                         CLR     R2              ;CLEAR THE LINE COUNTER.
6812 034470  010177  145542        25$:     MOV     R1,@CSRA        ;SET UP THE LINE NUMBER AND INTERRUPT ENABLE
6813                                                                 ;BITS IN THE CSR.
6814 034474  112777  000000  145542         MOVB    #0,@FDATA       ;SEND A NULL CHARACTER TO THE OUTPUT FIFO.
6815 034502  005201                         INC     R1              ;NEXT CSR CONTENTS.
6816 034504  005202                         INC     R2              ;NEXT LINE.
6817 034506  020227  000020                 CMP     R2,#NUMLNS      ;IF ALL LINES HAVE NOT BEEN SERVICED THEN
6818 034512  002766                         BLT     25$             ;BRANCH.
6819                                    ;+
6820                                    ; DELAY 250 MILLI-SECONDS TO ALLOW INTERRUPTS TO OCCUR.
6821                                    ;-
6822 034514  012704  000372                 MOV     #250.,R4        ;SET UP FOR 250 MS DELAY.
6823 034520  004767  163334                 JSR     PC,DELAY        ;WAIT 250 MS.
6824                                    ;+
6825                                    ; VERIFY THAT TX INTERRUPTS OCCURRED.
6826                                    ;-
6827 034524  005767  145572                 TST     TXINTC          ;CHECK THE TX INTERRUPT COUNTER.
6828 034530  001010                         BNE     26$             ;SKIP THE FOLLOWING ERROR IF WE GOT TX INTS.
6829                                    ;+
6830                                    ; DETERMINE THE REASON THAT WE RECEIVED NO INTERRUPTS.
6831                                    ;-
6832 034532  012701  013722                 MOV     #EM2610,R1      ;SET UP MSG IN CASE "TX_ACTION IS SET".
6833 034536  005777  145474                 TST     @CSRA           ;CHECK THE DUT CSR.
6834 034542  100410                         BMI     28$             ;GO TO REPORT ERROR IF TX_ACTION IS SET.
6835 034544  012701  014014                 MOV     #EM2611,R1      ;SET UP "TX_ACTION NOT SET" MESSAGE.
6836 034550  000405                         BR      28$             ;GO AND REPORT THE ERROR.
6837                                    ;+
6838                                    ; CHECK TO VERIFY THAT TX_ACTION WAS SET FOR EACH INTERRUPT.
6839                                    ;-
6840 034552  005767  145546        26$:     TST     TXINTF          ;CHECK THE TX INTERRUPT FLAGS.
6841 034556  100012                         BPL     30$             ;SKIP ERROR IF TX_ACTION CLR FLAG IS CLEAR.
6842 034560  012701  013643                 MOV     #EM2609,R1      ;SET UP TX INT WITH "TX_ACTION CLR" MSG.
6843                                    ;+
6844                                    ; REPORT "TRANSMIT INTERRUPT TEST ERROR:...."
6845                                    ;-
6846 034564                        28$:     ERRDF   2607,EM2606,ER0503;             >>>>> ERROR #2607 <<<<<.
      034564  104455                                                                      TRAP    C$ERDF
      034566  005057                                                                     .WORD   2607
      034570  013437                                                                     .WORD   EM2606
```

```
      034572  016072                                                                .WORD   ER0503
 6847
 6848                                    ;+
 6849                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
 6850                                    ;-
 6851 034574  032767  000100  145416          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
 6852 034602  001411                        BEQ     32$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
 6853
 6854                                ;+
 6855                                ; VERIFY THAT NO RX INTERRUPTS HAVE BEEN GENERATED SO FAR IN THIS TEST.
 6856                                ;-
 6857 034604  016702  145500        30$:     MOV     RXINTC,R2       ;LOAD # OF RX INTERRUPTS FOR ER0504 RTN.
 6858 034610  001406                        BEQ     32$             ;SKIP ERROR IF NO RX INTERRUPTS.
 6859 034612  012701  007431                MOV     #EM0525,R1      ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
 6860                                    ;REPORT "RX INTERRUPTS(S) RECEIVED WITH RX INTERRUPTS DISABLED."
 6861 034616                                ERRDF   2608,EM2606,ER0504;             >>>>> ERROR #2608 <<<<<
      034616  104455                                                                TRAP    C$ERDF
      034620  005060                                                                .WORD   2608
      034622  013437                                                                .WORD   EM2606
      034624  016130                                                                .WORD   ER0504
 6862
 6863                                ;+
 6864                                ; DISABLE INTERRUPTS AND CLEAN OUT THE INTERRUPT VECTORS USED IN THIS TEST.
 6865 034626  005001                ;-
 6866 034630  004767  166074        32$:     CLR     R1              ;CLEAR BOTH TRANSMITTER
 6867 034634  004767  165306                JSR     PC,TXIE0        ; INTERRUPT ENABLE AND RECEIVER
 6868 034640                                JSR     PC,RXIE0        ; INTERRUPT ENABLE BITS IN THE DUT CSR.
      034640  012700  000300        34$:     SETPRI  #PRI06          ;DISABLE DEVICE INTERRUPTS.
      034644  104441                                                                MOV     #PRI06,R0
 6869 034646                                                                        TRAP    C$SPRI
      034646  016700  145354                CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
      034652  104436                                                                MOV     RXVECA,R0
 6870 034654                                                                        TRAP    C$CVEC
      034654  016700  145350                CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
      034660  104436                                                                MOV     TXVECA,R0
 6871                                                                               TRAP    C$CVEC
 6872 034662  005067  145414        60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 6873 034666                                SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      034666  012700  000340                                                        MOV     #PRI07,R0
      034672  104441                                                                TRAP    C$SPRI
 6874 034674                                ENDTST
      034674                                                            L10053:
      034674  104401                                                               TRAP    C$ETST
 6875
```

```
 6877                                   .SBTTL  HARDWARE TEST           - BRLEVA -
 6878                                   ;++ ************************************************************************
 6879                                   ;*                      - BR LEVEL TEST B -
 6880                                   ;*      THIS TEST VERIFIES THAT THE DEVICE UNDER TEST (DUT) WILL GENERATE
 6881                                   ;*      RECEPTION AND TRANSMISSION INTERRUPTS AT THE CORRECT BR LEVEL.
 6882                                   ;*      THIS TEST DOES NOT DEPEND ON THE USE OF THE SERIAL LINE TRANSMISSION
 6883                                   ;*      OR RECEPTION CAPABILITIES OF THE DUT.  THE LINES ARE PUT IN INTERNAL
 6884                                   ;*      LOOPBACK TO MINIMIZE ANY EXTERNAL EFFECTS THAT COULD BE CAUSED ON
 6885                                   ;*      DEVICES ATTACHED TO THE SERIAL LINES.
 6886                                   ;*
 6887                                   ;-- ************************************************************************
 6888
 6889 034676                                   BGNTST
      034676                                                                                   T23::
 6890 034676                                   SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
      034676  012700  000240                                                                MOV     #PRI05,R0
      034702  104441                                                                       TRAP    C$SPRI
 6891            000027                        TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 6892 034704  012767  000027  145406           MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (30)
 6893 034712  012767  177777  145362           MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 6894 034720  012767  000001  147034           MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 6895 034726  012767  005671  147030           MOV     #3001.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
 6896 034734  012767  014173  147024           MOV     #EM3001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
 6897 034742  005067  145510                   CLR     ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
 6898                                   ;+
 6899                                   ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
 6900                                   ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
 6901                                   ; THIS SUBROUTINE REPORTS ERRORS FROM >>>>> 3001 THRU 3002 <<<<<.
 6902                                   ;-
 6903 034746  004767  164340                   JSR     PC,RESETT       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
 6904 034752  103402                           BCS     2$              ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
 6905 034754  000167  000644                   JMP     60$             ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
 6906 034760  012767  005673  146776  2$:      MOV     #3003.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 3003.
 6907                                   ;+
 6908                                   ; ENABLE TRANSMITTERS ON ALL LINES.
 6909                                   ;-
 6910 034766  012705  177777            4$:      MOV     #MAPLNS,R5      ;PASS ACTIVE LINE BIT MAP.
 6911 034772  004767  165636                    JSR     PC,TXENBL       ;ENABLE TRANSMISSION ON ALL LINES.
 6912                                   ;+
 6913                                   ; GENERATE A TRANSMISSION INTERRUPT REQUEST.
 6914                                   ; PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING INTS.
 6915                                   ;-
 6916 034776                                   SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      034776  012700  000340                                                                MOV     #PRI07,R0
      035002  104441                                                                       TRAP    C$SPRI
 6917 035004                                   SETVEC  TXVECA,#TXINTR,#PRI07 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
      035004  012746  000340                                                                MOV     #PRI07,-(SP)
      035010  012746  024026                                                                MOV     #TXINTR,-(SP)
      035014  016746  145210                                                                MOV     TXVECA,-(SP)
      035020  012746  000003                                                                MOV     #3,-(SP)
      035024  104437                                                                        TRAP    C$SVEC
      035026  062706  000010                                                                ADD     #10,SP
 6918                                       ;+
 6919                                       ; SET UP DUT FOR TRANSMISSION INTERRUPTS:
 6920                                       ;  SET UP INTERNAL LOOPBACK.
 6921                                       ;  SET UP LINE PARAMETERS FOR TRANSMISSION.
 6922                                       ;-
```

```
 6923 035032  012705  177777                MOV    #MAPLNS,R5      ;PASS ACTIVE LINES BIT MASK.
 6924 035036  012700  000200                MOV    #200,R0         ;PASS INERT STATE, INTERNAL LOOPBACK.
 6925 035042  004767  166364                JSR    PC,WTWLNC       ;DISABLE RECPTION AND DMA, ETC. ON DUT.
 6926 035046  012700  156430                MOV    #156430,R0      ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
 6927 035052  004767  166404                JSR    PC,WTWLPR       ;WRITE INTO ALL LPR REGISTERS.
 6928                                  ;+
 6929                                  ; SEND A NULL CHAR TO EACH LINE.
 6930                                  ;-
 6931 035056  016701  145222                MOV    IESTAT,R1       ;SET UP THE STATE OF THE INTERRUPT ENABLE BITS.
 6932 035062  010177  145150          5$:   MOV    R1,@CSRA        ;SET UP THE LINE NUMBER AND INTERRUPT ENABLE
 6933                                                               ;BITS IN THE CSR.
 6934 035066  112777  000000  145150      MOVB   #0,@FDATA       ;SEND A NULL CHARACTER TO THE OUTPUT FIFO.
 6935 035074  005201                      INC    R1              ;NEXT LINE.
 6936 035076  020127  000020              CMP    R1,#NUMLNS      ;IF ALL LINES HAVE NOT BEEN SERVICED THEN
 6937 035102  002767                      BLT    5$              ;BRANCH.
 6938                                  ;+
 6939                                  ; DELAY 50 MS TO ALLOW TIME FOR THE INTERRUPT TO BE GENERATED.
 6940                                  ;-
 6941 035104  012704  000062                MOV    #50.,R4         ;PASS 50 MS TIME TO THE DELAY ROUTINE.
 6942 035110  004767  162744                JSR    PC,DELAY        ;DELAY 50 MILLI-SECONDS.
 6943                                  ;+
 6944                                  ; GENERATE A RECEPTION INTERRUPT REQUEST.
 6945                                  ;-
 6946 035114                                SETVEC  RXVECA,#RXBRRT,#PRIO7 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
      035114  012746  000340                                                    MOV    #PRIO7,-(SP)
      035120  012746  023636                                                    MOV    #RXBRRT,-(SP)
      035124  016746  145076                                                    MOV    RXVECA,-(SP)
      035130  012746  000003                                                    MOV    #3,-(SP)
      035134  104437                                                           TRAP   C$SVEC
      035136  062706  000010                                                    ADD    #10,SP
 6947                                  ;+
 6948                                  ; SET UP FOR THE LOOP WHICH TESTS THE INTERRUPT BR LEVELS.
 6949                                  ;-
 6950 035142  012705  000340                MOV    #340,R5         ;SET UP THE PRIORITY LEVEL TO 7.
 6951 035146  005003                        CLR    R3              ;CLEAR THE RX PRIORITY STORE AND FLAGS.
 6952 035150  005002                        CLR    R2              ;CLEAR THE TX PRIORITY STORE AND FLAGS.
 6953                                  ;+
 6954                                  ; ENABLE TX AND RX INTERRUPTS.
 6955                                  ; PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING THE INTERRUPTS.
 6956                                  ;-
 6957 035152  004767  165030                JSR    PC,RXIE1        ;ENABLE RECEIVER INTERRUPTS.
 6958 035156  004767  165606                JSR    PC,TXIE1        ;ENABLE TRANSMITTER INTERRUPTS.
 6959                                  ;+
 6960                                  ; LOOP, LOWERING THE PROCESSOR PRIORITY UNTIL THE DUT INTERRUPTS ON RX AND TX.
 6961                                  ;-
 6962 035162  005067  145134          6$:   CLR    TXINTC          ;CLEAR THE TX INTERRUPT COUNTER.
 6963 035166  005067  145132                CLR    TXINTF          ;CLEAR THE TX INTERRUPT FLAGS.
 6964 035172  005067  145112                CLR    RXINTC          ;CLEAR THE RX INTERRUPT COUNTER.
 6965 035176  005067  145110                CLR    RXINTF          ;CLEAR THE RX INTERRUPT FLAGS.
 6966 035202                                SETPRI  R5              ;SET PROCESSOR PRIORITY TO THE SELECTED VALUE.
      035202  010500                                                   MOV    R5,R0
      035204  104441                                                   TRAP   C$SPRI
 6967 035206  012704  000001                MOV    #1,R4           ;PASS 1 MS COUNT TO THE DELAY ROUTINE.
 6968 035212  004767  162642                JSR    PC,DELAY        ;DELAY 1 MS TO ALLOW INTERRUPTS TO OCCUR.
 6969                                  ;+
 6970                                  ; DETERMINE IF ANY RX DUT INTERRUPTS OCCURRED.
 6971                                  ; LOG THE PROCESSOR PRIORITY FOR THE RX INTERRUPT IF FIRST RX INT.
```

```
                        ;-
035216  005767  145066          TST     RXINTC          ;CHECK THE RECEIVE INTERRUPT COUNTER.
035222  001412                  BEQ     8$              ;SKIP THE PRIORITY LOG IF NO RX INT OCCURRED.
                                ;+
                                ; IF THIS IS THE FIRST RX INTERRUPT, LOG THE PRIORITY.
                                ;-
035224  005703                   TST     R3              ;CHECK THE RX PRIORITY STORE AND FLAGS.
035226  001010                  BNE     8$              ;GOTO TEST FOR TX INTS IF NOT THE FIRST RX INT.
035230  010503                   MOV     R5,R3           ;LOG THE PRESENT PRIORITY IN THE RX PRIO STORE.
035232  052703  100000           BIS     #BIT15,R3       ;SET THE RX INT HAS OCCURRED FLAG.
035236  016700  145050           MOV     RXINTF,R0       ;GET THE RX INTERRUPT ROUTINE FLAGS.
035242  042700  137777           BIC     #137777,R0      ;CLEAR ALL BUT THE TX INT ERROR FLAG.
035246  050003                   BIS     R0,R3           ;IF TX INT ERROR, SET BIT 14 OF THE PRIO FLAGS.
                                ;+
                                ; DETERMINE IF ANY TX DUT INTERRUPTS HAVE OCCURRED.
                                ; LOG THE PRESENT PROCESSOR PRIORITY IF THIS IS THE FIRST TX INTERRUPT.
                                ;-
035250  005767  145046  8$:      TST     TXINTC          ;CHECK THE TRANSMIT INTERRUPT COUNTER.
035254  001405                  BEQ     10$             ;SKIP THE PRIORITY LOG IF NO TX INT OCCURRED.
                                ;+
                                ; IF THIS IS THE FIRST TX INTERRUPT, LOG THE PRIORITY.
                                ;-
035256  005702                   TST     R2              ;CHECK THE TX PRIORITY STORE AND FLAGS.
035260  100403                  BMI     10$             ;SKIP THE LOGGING IF NOT FIRST TX INTERRUPT.
035262  010502                   MOV     R5,R2           ;LOG THE PRESENT PRIORITY IN THE TX PRIO STORE.
035264  052702  100000           BIS     #BIT15,R2       ;SET THE TX INT HAS OCCURRED FLAG.
                                ;+
                                ; SELECT NEXT PROCESSOR PRIORITY.
                                ; TEST FOR BOTH RX AND TX INTERRUPTS HAVING OCCURRED, LOOP IF NOT.
                                ;-
035270  162705  000040  10$:      SUB     #40,R5          ;DECREMENT PRIORITY LEVEL BY ONE.
035274  002402                  BLT     12$             ;GOTO CHECK FOR ERRORS IF BELOW PRIORITY ZERO.
035276  030203                   BIT     R2,R3           ;AND PRIO FLAGS TOGETHER, ALTER NONE OF THEM.
035300  100330                  BPL     6$              ;LOOP IF RX AND TX INTS HAVEN'T BOTH OCCURRED.
                                ;+
                                ; DISABLE INTERRUPTS AND CLEAR INTERRUPT VECTORS.
                                ;-
035302                  12$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
035302  012700  000340                                                    MOV     #PRI07,R0
035306  104441                                                            TRAP    C$SPRI
035310                          CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
035310  016700  144712                                                    MOV     RXVECA,R0
035314  104436                                                            TRAP    C$CVEC
035316                          CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
035316  016700  144706                                                    MOV     TXVECA,R0
035322  104436                                                            TRAP    C$CVEC
                                ;+
                                ; VERIFY THAT RX AND TX INTERRUPTS OCCURRED,
                                ; AT THE PROPER BR LEVEL, AND
                                ; IN THE PROPER ORDER.
                                ;    DETERMINE IF TX INTERRUPT OCCURRED.
                                ;-
035324  005702                    TST     R2              ;DETERMINE WHETHER TX INT OCCURRED OR NOT.
035326  100420                   BMI     16$             ;SKIP THESE ERRORS IF TX INT OCCURRED.
                                ;+
                                ; DETERMINE REASON THAT NO TX INT OCCURRED.
                                ;-
```

```
7023 035330  012701  013722                    MOV     #EM2610,R1      ;SELECT "NO TX INT FROM TX.ACTION" MESSAGE.
7024 035334  005777  144676                    TST     @CSRA           ;CHECK THE TX.ACTION BIT OF THE DUT CSR.
7025 035340  100402                          BMI     14$             ;SKIP TX.ACTION CLR MSG SELECTION IF IT IS SET.
7026 035342  012701  014014                    MOV     #EM2611,R1      ;SELECT "TX.ACTION CLEAR AFTER CHARS SENT" MSG.
7027                                     ;REPORT "INTERRUPT BR LEVEL TEST ERROR:"
7028 035346                              14$:    ERRDF   3003,EM3001,ER0503;             >>>>> ERROR #3003 <<<<<.
     035346  104455                                                                              TRAP    C$ERDF
     035350  005673                                                                              .WORD   3003
     035352  014173                                                                              .WORD   EM3001
     035354  016072                                                                              .WORD   ER0503
7029
7030                                         ;+
7031                                         ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7032                                         ;-
7033 035356  032767  000100  144634              BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
7034 035364  001513                          BEQ     26$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7035
7036 035366  000427                          BR      18$             ;SKIP THE BR LEVEL CHECK, NO TX INT OCCURRED.
7037                                         ;+
7038                                         ; VERIFY THAT THE TX INTERRUPT WAS AT THE PROPER BR LEVEL.
7039                                         ;-
7040 035370  010204                      16$:    MOV     R2,R4           ;CALCULATE THE BR LEVEL
7041 035372  042704  177400                    BIC     #177400,R4      ; THAT THE TRANMSIT
7042 035376  006204                          ASR     R4              ; INTERRUPT WAS
7043 035400  006204                          ASR     R4              ; REQUESTED AT, WHICH
7044 035402  006204                          ASR     R4              ; IS ONE GREATER THAN
7045 035404  006204                          ASR     R4              ; THE PROCESSOR PRIORITY
7046 035406  006204                          ASR     R4              ; LEVEL AT WHICH THE
7047 035410  005204                          INC     R4              ; TRANSMIT INTERRUPT OCCURRED.
7048 035412  116705  144616                    MOVB    BRLEVL,R5       ;GET THE EXPECTED INTERRUPT BR LEVEL.
7049 035416  120405                          CMPB    R4,R5           ;COMPARE THE INTERRUPT BR LEVEL WITH EXPECTED.
7050 035420  001412                          BEQ     18$             ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
7051                                     ;REPORT "TX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
7052 035422  012701  014326                    MOV     #EM3003,R1      ;SELECT THE ERROR MESSAGE FOR THE ERROR CALL.
7053 035426                                  ERRDF   3004,EM3001,ER3001;             >>>>> ERROR #3004 <<<<<.
     035426  104455                                                                              TRAP    C$ERDF
     035430  005674                                                                              .WORD   3004
     035432  014173                                                                              .WORD   EM3001
     035434  016554                                                                              .WORD   ER3001
7054                                         ;+
7055                                         ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7056                                         ;-
7057 035436  032767  000100  144554              BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
7058 035444  001463                          BEQ     26$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7059
7060                                         ;+
7061                                         ; DETERMINE IF RX INTERRUPT OCCURRED.
7062                                         ;-
7063 035446  005703                      18$:    TST     R3              ;CHECK THE RX INT OCCURRED FLAG.
7064 035450  100421                          BMI     22$             ;SKIP THESE ERRORS IF RX INT OCCURRED.
7065                                         ;+
7066                                         ; DETERMINE REASON THAT NO RX INT OCCURRED.
7067                                         ;-
7068 035452  012701  014077                    MOV     #EM2612,R1      ;SELECT "NO RX INT FROM RX.DATA.AVAIL" MSG.
7069 035456  032777  000200  144552              BIT     #BIT7,@CSRA     ;CHECK THE RX.DATA.AVAIL BIT OF THE DUT CSR.
7070 035464  001002                          BNE     20$             ;SKIP RX.DATA.AVAIL CLR MSG IF BIT IS SET.
7071 035466  012701  014232                    MOV     #EM3002,R1      ;SELECT "NO RX.DATA.AVAIL AFTER RESET" MSG.
```

```
7072                                    ;REPORT "INTERRUPT BR LEVEL TEST ERROR:"
7073 035472                        20$:  ERRDF   3005,EM3001,ER0503;          >>>>> ERROR #3005 <<<<<.
     035472  104455                                                                 TRAP    C$ERDF
     035474  005675                                                                 .WORD   3005
     035476  014173                                                                 .WORD   EM3001
     035500  016072                                                                 .WORD   ER0503
7074                                    ;+
7075                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7076                                    ;-
7077 035502  032767  000100  144510        BIT    #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
7078 035510  001441                        BEQ    26$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7079
7080 035512  000427                        BR     24$             ;SKIP THE BR CHECK IF NO RX INT OCCURRED.
7081                                    ;+
7082                                    ; VERIFY THAT THE RX INTERRUPT WAS AT THE PROPER BR LEVEL.
7083                                    ;-
7084 035514  010304                 22$:   MOV    R3,R4           ;CALCULATE THE BR LEVEL
7085 035516  042704  177400               BIC    #177400,R4      ; THAT THE RECEIVE
7086 035522  006204                        ASR    R4              ; INTERRUPT WAS
7087 035524  006204                        ASR    R4              ; REQUESTED AT, WHICH
7088 035526  006204                        ASR    R4              ; IS ONE GREATER THAN
7089 035530  006204                        ASR    R4              ; THE PROCESSOR PRIORITY
7090 035532  006204                        ASR    R4              ; LEVEL AT WHICH THE
7091 035534  005204                        INC    R4              ; RECEIVE INTERRUPT OCCURRED.
7092 035536  116705  144472               MOVB   BRLEVL,R5       ;GET THE EXPECTED INTERRUPT BR LEVEL.
7093 035542  120405                        CMPB   R4,R5           ;COMARE THE INTERRUPT BR LEVEL WITH EXPECTED.
7094 035544  001412                        BEQ    24$             ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
7095                                    ;REPORT "RX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
7096 035546  012701  014402               MOV    #EM3004,R1      ;SELECT ERROR MESSAGE FOR THE ERROR CALL.
7097 035552                               ERRDF  3006,EM3001,ER3001;          >>>>> ERROR #3006 <<<<<.
     035552  104455                                                                 TRAP    C$ERDF
     035554  005676                                                                 .WORD   3006
     035556  014173                                                                 .WORD   EM3001
     035560  016554                                                                 .WORD   ER3001
7098                                    ;+
7099                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7100                                    ;-
7101 035562  032767  000100  144430        BIT    #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
7102 035570  001411                        BEQ    26$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7103
7104                                    ;+
7105                                    ; TEST FOR INTERRUPTS OCCURING IN THE PROPER ORDER.
7106                                    ;-
7107 035572  032703  040000         24$:   BIT    #BIT14,R3       ;CHECK THE IMPROPER INT ORDER ERROR FLAG.
7108 035576  001406                        BEQ    26$             ;SKIP ERROR REPORT IF ERROR DID NOT OCCUR.
7109                                    ;REPORT "TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS RX INT."
7110 035600  012701  014456               MOV    #EM3005,R1      ;SELECT THE ERROR MESSAGE FOR INDIRECT PRINT.
7111 035604                               ERRDF  3007,EM3001,ER0503;          >>>>> ERROR #3007 <<<<<.
     035604  104455                                                                 TRAP    C$ERDF
     035606  005677                                                                 .WORD   3007
     035610  014173                                                                 .WORD   EM3001
     035612  016072                                                                 .WORD   ER0503
7112                                    ;+
7113                                    ; CLEAN UP, EXIT THE TEST.
7114                                    ;-
7115 035614  004767  165110         26$:   JSR    PC,TXIE0        ;CLEAR TRANSMITTER INTERRUPTS.
7116 035620  004767  164322               JSR    PC,RXIE0        ;CLEAR RECEIVER INTERRUPTS.
```

```
   7117 035624  005067  144452          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
   7118 035630                                  SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
        035630  012700  000340                                                  MOV     #PRIO7,R0
        035634  104441                                                          TRAP    C$SPRI
   7119 035636                                  ENDTST
        035636                                                          L10054:
        035636  104401                                                          TRAP    C$ETST
   7120
```

```
                                .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.
                                ;*
                                ;-- ******************************************************************************
035640                                  BGNTST
035640                                                                          T24::
        000030                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
035640  012767  000030  144452   MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (93)
035646  012767  177777  144426   MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
035654  016702  144640           MOV     BMPCQP,R2      ;GET THE CONTENTS OF THE POINTER.
035660  012703  002522           MOV     #BMPCQB,R3     ;GET THE START ADDRESS OF THE QUEUE.
035664  020203                   CMP     R2,R3          ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
035666  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"

035670  012701  015370           MOV     #EM9304,R1     ;PASS THE FIRST MESSAGE TO BE REORTED.
035674                           ERRDF   9301,EM9301,ER9301 ;           >>>>> ERROR #9301 <<<<<.
035674  104455                                                                  TRAP    C$ERDF
035676  022125                                                                  .WORD   9301
035700  015253                                                                  .WORD   EM9301
035702  017216                                                                  .WORD   ER9301

035704  012767  002522  144606           MOV     #BMPCQB,BMPCQP ;SET POINTER BACK TO THE BEGINING OF THE QUE.

035712  005067  144364   60$:    CLR     CTRLCF         ;INDICATE THAT WE ARE NOT WITHIN A TEST.
035716                           ENDTST
035716                                                                  L10055:
035716  104401                                                                  TRAP    C$ETST
```

```
7154                                    ;*****************************************************************************************
7155                                    ;
7156                                    ;               FVTB.HWQ
7157                                    ;
7158                                    ;*****************************************************************************************
7160
7161
7162                                    .SBTTL  HARDWARE PARAMETER CODING SECTION
7163
7164
7165
7166                                    ;++
7167                                    ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
7168                                    ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7169                                    ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7170                                    ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
7171                                    ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
7172                                    ; WITH THE OPERATOR.
7173                                    ;--
7174
7175 035720                                    BGNHRD
      035720  000022                                                        .WORD L10056-L$HARD/2
      035722                                                          L$HARD::
7176
7186                                    ;DEVICE CSR ADDRESS QUESTION:
7187 035722                                    GPRMA   HWPTQ1,0,0,160000,177776,YES
      035722  000031                                                        .WORD   T$CODE
      035724  035766                                                        .WORD   HWPTQ1
      035726  160000                                                        .WORD   T$LOLIM
      035730  177776                                                        .WORD   T$HILIM
7188
7189 035732                            ;DEVICE INTERRUPT VECTOR QUESTION:
                                               GPRMA   HWPTQ2,2,0,40,776,YES
      035732  001031                                                        .WORD   T$CODE
      035734  036004                                                        .WORD   HWPTQ2
      035736  000040                                                        .WORD   T$LOLIM
      035740  000776                                                        .WORD   T$HILIM
7190
7191 035742                            ;ACTIVE LINES BIT MAP QUESTION:
                                               GPRMD   HWPTQ3,4,0,MAPLNS,0,MAPLNS,YES
      035742  002032                                                        .WORD   T$CODE
      035744  036037                                                        .WORD   HWPTQ3
      035746  177777                                                        .WORD   MAPLNS
      035750  000000                                                        .WORD   T$LOLIM
      035752  177777                                                        .WORD   T$HILIM
7192
7193 035754                            ;INTERRUPT BR LEVEL QUESTION:
                                               GPRMD   HWPTQ4,6,0,377,0,6,YES
      035754  003032                                                        .WORD   T$CODE
      035756  036065                                                        .WORD   HWPTQ4
      035760  000377                                                        .WORD   377
      035762  000000                                                        .WORD   T$LOLIM
      035764  000006                                                        .WORD   T$HILIM
7194
7195
7196 035766                                    ENDHRD
                                                                            .EVEN
      035766                                                          L10056:
7197
7204
```

```
7205 035766  103  123  122        HWPTQ1: .ASCIZ  /CSR ADDRESS: /
     035771  040  101  104
     035774  104  122  105
     035777  123  123  072
     036002  040  000
7206 036004  111  116  124        HWPTQ2: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
     036007  105  122  122
     036012  125  120  124
     036015  040  126  105
     036020  103  124  117
     036023  122  040  101
     036026  104  104  122
     036031  105  123  123
     036034  072  040  000
7207 036037  101  103  124        HWPTQ3: .ASCIZ  /ACTIVE LINE BIT MAP: /
     036042  111  126  105
     036045  040  114  111
     036050  116  105  040
     036053  102  111  124
     036056  040  115  101
     036061  120  072  040
     036064  000
7208 036065  111  116  124        HWPTQ4: .ASCIZ  /INTERRUPT BR LEVEL: /
     036070  105  122  122
     036073  125  120  124
     036076  040  102  122
     036101  040  114  105
     036104  126  105  114
     036107  072  040  000
7209
7210                             .EVEN
```

```
                        ;*******************************************************************************
                        ;
                        ;               FVTA.SWQ
                        ;
                        ;*******************************************************************************


                        .SBTTL  SOFTWARE PARAMETER CODING SECTION

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

036112                          BGNSFT
036112  000017                                                  .WORD L10057-L$SOFT/2
036114                                                  L$SOFT::

                        ;UNIT NUMBER PRINTOUT QUESTION:
036114                          GPRML   SWPTQ1,0,20,YES
036114  000130                                                  .WORD   T$CODE
036116  036152                                                  .WORD   SWPTQ1
036120  000020                                                  .WORD   20

                        ;ROM VERSION NUMBER PRINTOUT ON FIRST PASS QUESTION:
036122                          GPRML   SWPTQ2,0,1,YES
036122  000130                                                  .WORD   T$CODE
036124  036226                                                  .WORD   SWPTQ2
036126  000001                                                  .WORD   1

                        ;EXTENDED ERROR REPORTING QUESTION:
036130                          GPRML   SWPTQ3,0,100,YES
036130  000130                                                  .WORD   T$CODE
036132  036277                                                  .WORD   SWPTQ3
036134  000100                                                  .WORD   100

                          ;+
                          ; IF EXTENDED ERROR REPORTING IS NOT REQUIRED THEN SKIP THE NEXT QUESTION.
                          ;+
036136                          XFERF   ENDD
036136  006044                                                  .WORD   T$CODE

                        ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
036140                          GPRMD   SWPTQ4,2,D,177777,0,177777,YES
036140  001052                                                  .WORD   T$CODE
036142  036332                                                  .WORD   SWPTQ4
036144  177777                                                  .WORD   177777
036146  000000                                                  .WORD   T$LOLIM
036150  177777                                                  .WORD   T$HILIM

                        .EVEN

036152  ENDD:   ENDSFT
                                                                .EVEN
036152                                                  L10057:
```

```
 7266 036152  122  105  120        SWPTQ1: .ASCIZ  /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /
      036155  117  122  124
      036160  040  125  116
      036163  111  124  040
      036166  116  125  115
      036171  102  105  122
      036174  040  101  123
      036177  040  105  101
      036202  103  110  040
      036205  125  116  111
      036210  124  040  111
      036213  123  040  124
      036216  105  123  124
      036221  105  104  072
      036224  040  000
 7267 036226  122  117  115        SWPTQ2: .ASCIZ  /ROM VERSION PRINTOUT ON THE FIRST PASS: /
      036231  040  126  105
      036234  122  123  111
      036237  117  116  040
      036242  120  122  111
      036245  116  124  117
      036250  125  124  040
      036253  117  116  040
      036256  124  110  105
      036261  040  106  111
      036264  122  123  124
      036267  040  120  101
      036272  123  123  072
      036275  040  000
 7268 036277  105  130  124        SWPTQ3: .ASCIZ  /EXTENDED ERROR REPORTING: /
      036302  105  116  104
      036305  105  104  040
      036310  105  122  122
      036313  117  122  040
      036316  122  105  120
      036321  117  122  124
      036324  111  116  107
      036327  072  040  000
 7269 036332  116  125  115        SWPTQ4: .ASCIZ  /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /
      036335  102  105  122
      036340  040  117  106
      036343  040  111  116
      036346  104  111  126
      036351  111  104  125
      036354  101  114  040
      036357  104  101  124
      036362  101  040  105
      036365  122  122  117
      036370  122  123  040
      036373  124  117  040
      036376  122  105  120
      036401  117  122  124
      036404  040  117  116
      036407  040  101  040
      036412  114  111  116
      036415  105  072  040
      036420  000
```

```
7270                                    .EVEN
```

```
 7272
 7273                           ;*****************************************************************************************
 7274                           ;
 7275                           ;               FVTSKL6.P11
 7276                           ;
 7277                           ;*****************************************************************************************
 7278
 7279
 7280
 7281 036422                   $PATCH::
 7282 036422                           .BLKW   24
 7283
 7290
 7291
 7292
 7293
 7294 036472                           LASTAD
                                                                                        .EVEN
      036472  000000                                                                    .WORD   0
      036474  000000                                                                    .WORD   0
      036476                   L$LAST::
 7295 036476                           ENDMOD
 7296
 7297
 7298
 7299
 7300
 7301
 7302
 7303          000001                   .END
```

SYMBOL TABLE

```
ACTLNS  002224 G        C$AU  = 000052        DROP    025002          EM0902  011002 G        EM9024  015235 G
ADR   = 000020 G        C$AUTO= 000061        DR00MG  005470 G        EM1001  011036 G        EM9301  015253 G
ADRPTR  017720 G        C$BRK = 000022        DR02MG  005474 G        EM1002  011071 G        EM9302  015273 G
ALTFLD  017410 G        C$BSEG= 000004        DR04MG  005501 G        EM1003  011156 G        EM9303  015323 G
ASSEMB= 000010          C$BSUB= 000002        DR06MG  005505 G        EM1101  011234 G        EM9304  015370 G
BCOUNT  002346 G        C$CEFG= 000045        DR10MG  005523 G        EM1201  011307 G        ENDD    036152
BITTBL  002404 G        C$CLCK= 000062        DR12MG  005532 G        EM1202  011335 G        ENDETB  003722 G
BIT0  = 000001 G        C$CLEA= 000012        DR14MG  005543 G        EM1203  011421 G        ENDIT   024722
BIT00 = 000001 G        C$CLOS= 000035        DR16MG  005554 G        EM1204  011477 G        ERCNTB  002460 G
BIT01 = 000002 G        C$CLP1= 000006        EDROP   025060          EM1205  011533 G        ERLTBL  002722 G
BIT02 = 000004 G        C$CVEC= 000036        EF.CON= 000036 G        EM1301  011557 G        ERRBLK  003770 G
BIT03 = 000010 G        C$DCLN= 000044        EF.NEW= 000035 G        EM1302  011612 G        ERRMSG  003766 G
BIT04 = 000020 G        C$DODU= 000051        EF.PWR= 000034 G        EM1401  011651 G        ERRNBR  003764 G
BIT05 = 000040 G        C$DRPT= 000024        EF.RES= 000037 G        EM1402  011710 G        ERRTYP  003762 G
BIT06 = 000100 G        C$DU  = 000053        EF.STA= 000040 G        EM1403  011776 G        ERSMRF  002456 G
BIT07 = 000200 G        C$EDIT= 000003        EF0503  004121 G        EM1404  012051 G        ER0101  015444 G
BIT08 = 000400 G        C$ERDF= 000055        EF0505  004126 G        EM1405  012123 G        ER0201  015776 G
BIT09 = 001000 G        C$ERHR= 000056        EF1401  004201 G        EM1406  012136 G        ER0503  016072 G
BIT1  = 000002 G        C$ERRO= 000060        EF1402  004303 G        EM1407  012151 G        ER0504  016130 G
BIT10 = 002000 G        C$ERSF= 000054        EF1601  004340 G        EM1408  012163 G        ER1401  016210 G
BIT11 = 004000 G        C$ERSO= 000057        EF1602  004372 G        EM1601  012171 G        ER1601  016352 G
BIT12 = 010000 G        C$ESCA= 000010        EF1603  004434 G        EM1604  012254 G        ER1603  016462 G
BIT13 = 020000 G        C$ESEG= 000005        EF1604  004476 G        EM1701  012330 G        ER3001  016554 G
BIT14 = 040000 G        C$ESUB= 000003        EF3001  004573 G        EM1801  012413 G        ER9004  016656 G
BIT15 = 100000 G        C$ETST= 000001        EF3002  004642 G        EM1901  012467 G        ER9007  016772 G
BIT2  = 000004 G        C$EXIT= 000032        EF9006  004711 G        EM2001  012552 G        ER9008  017062 G
BIT3  = 000010 G        C$GETB= 000026        EF9010  004735 G        EM2002  012622 G        ER9101  017156 G
BIT4  = 000020 G        C$GETW= 000027        EF9016  005034 G        EM2301  012675 G        ER9301  017216 G
BIT5  = 000040 G        C$GMAN= 000043        EF9017  005131 G        EM2302  012742 G        EVL   = 000004 G
BIT6  = 000100 G        C$GPHR= 000042        EF9018  005205 G        EM2401  013000 G        EXOERR  002300 G
BIT7  = 000200 G        C$GPLO= 000030        EF9301  005312 G        EM2601  013043 G        E$END = 002100
BIT8  = 000400 G        C$GPRI= 000040        EF9302  005370 G        EM2602  013101 G        E$LOAD= 000035
BIT9  = 001000 G        C$INIT= 000011        EM0101  020314 G        EM2603  013172 G        FDATA   002244 G
BMPCQB  002522 G        C$INLP= 000020        EM0102  020400 G        EM2604  013260 G        FSLSA   002244 G
BMPCQE  002722 G        C$MANI= 000050        EM0103  005564 G        EM2605  013354 G        FSLSO = 000006 G
BMPCQP  002520 G        C$MEM = 000031        EM0201  005632 G        EM2606  013437 G        F$AU  = 000015
BOE   = 000400 G        C$MSG = 000023        EM0202  005665 G        EM2607  013476 G        F$AUTO= 000020
BRLEVL  002234 G        C$OPEN= 000034        EM0203  006040 G        EM2608  013572 G        F$BGN = 000040
BUFBAS  002722 G        C$PNTB= 000014        EM0204  006203 G        EM2609  013643 G        F$CLEA= 000007
BUFEND  003722 G        C$PNTF= 000017        EM0301  006362 G        EM2610  013722 G        F$DU  = 000016
BUFMID  003322 G        C$PNTS= 000016        EM0302  006434 G        EM2611  014014 G        F$END = 000041
BUFPTR  002276 G        C$PNTX= 000015        EM0303  006574 G        EM2612  014077 G        F$HARD= 000004
BUF3QT  003522 G        C$QIO = 000377        EM0401  006753 G        EM3001  014173 G        F$HW  = 000013
CACHRX  023512 G        C$RDBU= 000007        EM0402  007031 G        EM3002  014232 G        F$INIT= 000006
CACHTX  023540 G        C$REFG= 000047        EM0501  007201 G        EM3003  014326 G        F$JMP = 000050
CALMSL  017462 G        C$RESE= 000033        EM0502  007255 G        EM3004  014402 G        F$MOD = 000000
CKTRAP  017706 G        C$REVI= 000003        EM0525  007431 G        EM3005  014456 G        F$MSG = 000011
CLKBRL  002332 G        C$RFLA= 000021        EM0526  007521 G        EM3101  014550 G        F$PROT= 000021
CLKCSR  002330 G        C$RPT = 000025        EM0601  007611 G        EM3102  014613 G        F$PWR = 000017
CLKHRZ  002336 G        C$SEFG= 000046        EM0602  007653 G        EM9010  014672 G        F$RPT = 000012
CLKINT  023566 G        C$SPRI= 000041        EM0603  010033 G        EM9014  014716 G        F$SEG = 000003
CLKVEC  002334 G        C$SVEC= 000037        EM0701  010216 G        EM9017  015012 G        F$SOFT= 000005
CLNRST  017736 G        C$TPRI= 000013        EM0702  010261 G        EM9018  015123 G        F$SRV = 000010
CLR16W  017760 G        DELAY   020060 G      EM0703  010441 G        EM9019  015133 G        F$SUB = 000002
CNTERR  020002 G        DFPTBL  002206 G      EM0801  010624 G        EM9020  015150 G        F$SW  = 000014
CSRA    002236 G        DIAGMC= 000000        EM0802  010672 G        EM9022  015174 G        F$TEST= 000001
CTRLCF  002302 G        DRADRT  002236 G      EM0901  010742 G        EM9023  015213 G        GETPRM  024520
```

SYMBOL TABLE

```
GPRS0B  002444 G    L$ACP   002110 G    L10003  016070    NEWPAS  024500      SVCINS= 000001
G$CNTO= 000200      L$APT   002036 G    L10004  016126    NEWRES  024472      SVCSUB= 000001
G$DELM= 000372      L$AU    025066 G    L10005  016206    NEWSTA  024162      SVCTAG= 000001
G$DISP= 000003      L$AUT   002070 G    L10006  016350    NUMLNS= 000020 G    SVCTST= 000001
G$EXCP= 000400      L$AUTO  024736 G    L10007  016460    OOPS    020250 G    SWAP0   022356 G
G$HILI= 000002      L$CCP   002106 G    L10010  016552    OPTION  002220 G    SWPTQ1  036152
G$LOLI= 000001      L$CLEA  024740 G    L10011  016654    O$APTS= 000000      SWPTQ2  036226
G$NO  = 000000      L$CO    002032 G    L10012  016770    O$AU  = 000000      SWPTQ3  036277
G$OFFS= 000400      L$DEPO  002011 G    L10013  017060    O$BGNR= 000001      SWPTQ4  036332
G$OFSI= 000376      L$DESC  004042 G    L10014  017154    O$BGNS= 000001      S$LSYM= 010000
G$PRMA= 000001      L$DESP  002076 G    L10015  017214    O$DU  = 000001      TIMER1  002340 G
G$PRMD= 000002      L.$DEVP 002060 G    L10016  017406    O$ERRT= 000001      TIMER2  002342 G
G$PRML= 000000      L$DISP  002124 G    L10017  024106    O$GNSW= 000001      TIMER3  002344 G
G$RADA= 000140      L$DLY   002116 G    L10021  024734    O$POIN= 000001      TNUM  = 000030 G
G$RADB= 000000      L$DTP   002040 G    L10022  024736    O$SETU= 000000      TP4FLG  002314 G
G$RADD= 000040      L$DTYP  002034 G    L10023  024754    PAR0A   002362 G    TP4RTN  024004 G
G$RADL= 000120      L$DU    024756 G    L10024  025064    PASCNT  002306 G    TP4VEC  002316 G
G$RADO= 000020      L$DUT   002072 G    L10025  025072    PCSLOT= 000016 G    TSABRT  022426 G
G$XFER= 000004      L$DVTY  004032 G    L10026  025354    PNT   = 001000 G    TSTNUM  002320 G
G$YES = 000010      L$EF    002052 G    L10027  025604    PREGRT  004014 G    TXAD1A  002250 G
HELP  = 000000      L$ENVI  002044 G    L10030  026050    PREG05  003772      TXAD2A  002252 G
HOE   = 100000 G    L$ERRT  003762 G    L10031  026246    PRI   = 002000 G    TXBFCA  002254 G
HWPTQ1  035766      L$ETP   002102 G    L10032  026440    PRI00 = 000000 G    TXBFCO= 000016 G
HWPTQ2  036004      L$EXP1  002046 G    L10033  026656    PRI01 = 000040 G    TXDSBL  022540 G
HWPTQ3  036037      L$EXP4  002064 G    L10034  027064    PRI02 = 000100 G    TXENBL  022634 G
HWPTQ4  036065      L$EXP5  002066 G    L10035  027272    PRI03 = 000140 G    TXIE0   022730 G
IBE   = 010000 G    L$HARD  035722 G    L10036  027466    PRI04 = 000200 G    TXIE1   022770 G
IDU   = 000040 G    L$HIME  002120 G    L10037  027702    PRI05 = 000240 G    TXINTC  002322 G
IER   = 020000 G    L$HPCP  002016 G    L10040  030130    PRI06 = 000300 G    TXINTF  002324 G
IESTAT  002304 G    L$HPTP  002022 G    L10041  030424    PRI07 = 000340 G    TXINTR  024026 G
ISR   = 000100 G    L$HW    002206 G    L10042  030712    PUFIFO  020476 G    TXVECA  002230 G
IXE   = 004000 G    L$ICP   002104 G    L10043  031312    RBUFA   002240 G    T$ARGC= 000003
I$AU  = 000041      L$INIT  024116 G    L10044  031560    RDPDR   020560 G    T$CODE= 001052
I$AUTO= 000041      L$LADP  002026 G    L10045  031750    REGTST  021032 G    T$ERRN= 022125
I$CLN = 000041      L$LAST  036476 G    L10046  032266    REPSMR  021264 G    T$EXCP= 000000
I$DU  = 000041      L$LOAD  002100 G    L10047  032530    RESETT  021312 G    T$FLAG= 000050
I$HRD = 000041      L$LUN   002074 G    L10050  032644    RMATBB  002256 G    T$GMAN= 000000
I$INIT= 000041      L$MREV  002050 G    L10051  033162    ROLDAP  021424 G    T$HILI= 177777
I$MOD = 000041      L$NAME  002000 G    L10052  033536    RSTRPT  021450 G    T$LAST= 000001
I$MSG = 000041      L$PRIO  002042 G    L10053  034674    RXBRRT  023636 G    T$LOLI= 000000
I$PROT= 000040      L$PROT  024110 G    L10054  035636    RXIE0   022146 G    T$LSYM= 010000
I$PTAB= 000041      L$PRT   002112 G    L10055  035716    RXIE1   022206 G    T$LTNO= 000030
I$PWR = 000041      L$REPP  002062 G    L10056  035766    RXINPT  023716 G    T$NEST= 177777
I$RPT = 000041      L$REV   002010 G    L10057  036152    RXINTC  002310 G    T$NS0 = 000000
I$SEG = 000041      L$RPT   024102 G    MAPLNS= 177777 G    RXINTF  002312 G    T$NS1 = 000005
I$SETU= 000041      L$SOFT  036114 G    MFUNIT  004070 G    RXTMA   002240 G    T$PTNU= 000000
I$SFT = 000041      L$SPC   002056 G    MMENAB  002360 G    RXVECA  002226 G    T$SAVL= 177777
I$SRV = 000041      L$SPCP  002020 G    MMPRES  002356 G    R0SLOT= 000002 G    T$SEGL= 177777
I$SUB = 000041      L$SPTP  002024 G    MMSR0   002354 G    R1SLOT= 000004 G    T$SUBN= 000000
I$TST = 000041      L$STA   002030 G    MSG1    015562 G    R2SLOT= 000006 G    T$TAGL= 177777
J$JMP = 000167      L$SW    002220 G    MSG2    015640 G    R3SLOT= 000010 G    T$TAGN= 010060
LNCTRA  002246 G    L$TEST  002114 G    MSG3    015717 G    R4SLOT= 000012 G    T$TEMP= 000000
LOE   = 040000 G    L$TIML  002014 G    MSLCNT  002352 G    R5SLOT= 000014 G    T$TEST= 000030
LOT   = 000010 G    L$UNIT  002012 G    MSLGET  020120 G    SAVBMP  022232 G    T$TSTM= 177777
LPCSLT= 000036 G    L10000  002216      MSLOOP  020234 G    SFPTBL  002220 G    T$TSTS= 000001
LPRA    002242 G    L10001  002224      MSTICK  002350 G    SKPSTS  022300 G    T$$AU = 010025
LPR0  = 000004 G    L10002  015560      NDERPT  002222 G    SVCGBL= 000000      T$$AUT= 010022
```

SYMBOL TABLE

```
T$$CLE= 010023          T$$TES= 010055          T18     032270 G        T5      026250 G        WDPDR   023224 G
T$$DU = 010024          T1      025074 G        T19     032532 G        T6      026442 G        WORD1   002326 G
T$$HAR= 010056          T10     027470 G        T2      025356 G        T7      026660 G        WTWLNC  023432 G
T$$HW = 010000          T11     027704 G        T20     032646 G        T8      027066 G        WTWLPR  023462 G
T$$INI= 010021          T12     030132 G        T21     033164 G        T9      027274 G        X$ALWA= 000000
T$$MSG= 010016          T13     030426 G        T22     033540 G        UAM   = 000200 G        X$FALS= 000040
T$$PRO= 010020          T14     030714 G        T23     034676 G        UNBTTB  002364 G        X$OFFS= 000400
T$$RPT= 010017          T15     031314 G        T24     035640 G        UNITN   002232 G        X$TRUE= 000020
T$$SOF= 010057          T16     031562 G        T3      025606 G        UNSDIV  023014 G        $PATCH  036422 G
T$$SW = 010001          T17     031752 G        T4      026052 G        WAIBIS  023150 G

. ABS.  036476      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  28661 WORDS  ( 112 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:04:00
$PARTA.BIN,PARTA.LST/-SP-SVC34R/ML,PARTA.P11
```