PROGRAM DOCUMENT

```
.REM &

                    IDENTIFICATION

PRODUCT CODE:       AC T794R MC
PRODUCT NAME:       CZDHUBO DHU 11 FUNC TST PART1
PRODUCT DATE:       3 MARCH 1984
MAINTAINER:         ENE - DIAGNOSTICS GROUP
AUTHOR:             ANTHONY HART
MODIFIED BY:        ANTHONY HART
```

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

ORIGINAL RELEASE:	15-DEC-83	ANTHONY HART
VERSION 80	3-MAR-84	ANTHONY HART

TWO NEW TESTS WERE INCLUDED IN THIS PART:

TEST 15 - CSR BIT 4 TEST.
TEST 25 - DIAGNOSTIC FIELD (BMP) TEST.

TABLE OF CONTENTS

1.0 GENERAL PROGRAM CONSIDERATIONS

1.1 PROGRAM ABSTRACT

CZDHUBO IS PART ONE OF THE DHU FUNCTIONAL VERIFICATION TEST. THIS PART OF THE TEST VERIFIES THE RESET, SELFTEST, REGISTER ACCESS,BMP CODE, 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 DHU11 FVT:

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

1.3 RELATED DOCUMENTS AND STANDARDS

- 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)
/FLAGS	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					
/FLAGS					
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.
BSE	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
****
)D/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 EVT.

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 0 (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.

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

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

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

UNIT 7
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE 0 (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.

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

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

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

2.8 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI/UNIBUS AND 50HZ (IF THERE IS A CLOCK) QUESTIONS. NOTE, NOT ALL VERSIONS OF XXDP+ ASK FOR THE CLOCK FREQUENCY
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:

```
C2DHU DVC FTL ERR 01603 ON UNIT 02 TST 16 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 ERROR WOULD BE REPORTED AS FOLLOWS:

```
C2DHU DVC FTL ERR 01603 ON UNIT 02 TST 16 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 (O).
    ACTUAL DATA:   000023 (O).
```

4.0 PERFORMANCE AND PROGRESS REPORTS

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

5.0 TEST SUMMARIES

THE FOLLOWING ARE INCLUDED WITHIN CZDHUB:

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. CSR BIT 4 TEST - VERIFIES THAT WHEN SET THIS BIT CAUSES THE SELFTEST TO LOOP, AND WHEN CLEARED THE SKIP SELFTEST CODES ARE RETURNED IN THE RXFIFO.

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

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

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

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

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

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

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

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

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

25. DIAGNOSTIC FIELD (BMP) TEST - VERIFIES THAT A REQUEST TO THE DUT TO REPORT BMP STATUS CODES IS COMPLIED WITH WITHIN THE SPECIFIED TIME. ALL ACTIVE LINES ARE TESTED.

26. REPORT BMP CODES TEST - THIS PSEUDO TEST REPORTS THE FIRST 32 BMP 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 CZDHUBO
CZDHUBO.BIN


DR>
CZDHU-B-0
DHU 11 FUNC TST PART1
UNIT IS DHU-11
RESTRT ADDR:   147670
DR>STA/PAS:1

CHANGE HW (L)   ? Y

# UNITS (D)  ?  2

UNIT 0
CSR ADDRESS:  (O)  160460 ? tZ

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

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


                                  .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,B,0,16,0,PRI07
     002000                                                                     L$NAME::
     002000     103                                                                         .ASCII /C/
     002001     132                                                                         .ASCII /Z/
     002002     104                                                                         .ASCII /D/
     002003     110                                                                         .ASCII /H/
     002004     125                                                                         .ASCII /U/
     002005     000                                                                         .BYTE   0
     002006     000                                                                         .BYTE   0
     002007     000                                                                         .BYTE   0
     002010                                                                     L$REV::
```

```
002010     102                                          .ASCII  /B/
002011                                        L$DEPO::
002011     060                                          .ASCII  /0/
002012                                        L$UNIT::
002012  000000                                          .WORD   0
002014                                        L$TIML::
002014  000016                                          .WORD   16
002016                                        L$HPCP::
002016  036700                                          .WORD   L$HARD
002020                                        L$SPCP::
002020  037072                                          .WORD   L$SOFT
002022                                        L$HPTP::
002022  002212                                          .WORD   L$HW
002024                                        L$SPTP::
002024  002224                                          .WORD   L$SW
002026                                        L$LADP::
002026  037454                                          .WORD   L$LAST
002030                                        L$STA::
002030  000000                                          .WORD   0
002032                                        L$CO::
002032  000000                                          .WORD   0
002034                                        L$DTYP::
002034  000000                                          .WORD   0
002036                                        L$APT::
002036  000000                                          .WORD   0
002040                                        L$DTP::
002040  002124                                          .WORD   L$DISPATCH
002042                                        L$PRIO::
002042  000340                                          .WORD   PRIO7
002044                                        L$ENVI::
002044  000000                                          .WORD   0
002046                                        L$EXP1::
002046  000000                                          .WORD   0
002050                                        L$MREV::
002050     003                                          .BYTE   C$REVISION
002051     003                                          .BYTE   C$EDIT
002052                                        L$EF::
002052  000000                                          .WORD   0
002054  000000                                          .WORD   0
002056                                        L$SPC::
002056  000000                                          .WORD   0
002060                                        L$DEVP::
002060  004036                                          .WORD   L$DVTYP
002062                                        L$REPP::
002062  004222                                          .WORD   L$RPT
002064                                        L$EXP4::
002064  000000                                          .WORD   0
002066                                        L$EXP5::
002066  000000                                          .WORD   0
002070                                        L$AUT::
002070  000000                                          .WORD   0
002072                                        L$DUT::
002072  025076                                          .WORD   L$DU
002074                                        L$LUN::
002074  000000                                          .WORD   0
002076                                        L$DESP::
002076  004046                                          .WORD   L$DESC
```

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.

DISPATCH TABLE

```
                        .SBTTL  DISPATCH TABLE

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

                                DISPATCH 26
002122                                                          .WORD   26
002122  000032
002124                                                  L$DISPATCH::
002124  025214                                                  .WORD   T1
002126  025476                                                 .WORD   T2
002130  025726                                                 .WORD   T3
002132  026172                                                 .WORD   T4
002134  026370                                                 .WORD   T5
002136  026562                                                 .WORD   T6
002140  027000                                                 .WORD   T7
002142  027206                                                 .WORD   T8
002144  027414                                                 .WORD   T9
002146  027610                                                 .WORD   T10
002150  030024                                                 .WORD   T11
002152  030252                                                 .WORD   T12
002154  030546                                                 .WORD   T13
002156  031034                                                 .WORD   T14
002160  031434                                                 .WORD   T15
002162  031776                                                 .WORD   T16
002164  032244                                                 .WORD   T17
002166  032434                                                 .WORD   T18
002170  032752                                                 .WORD   T19
002172  033214                                                 .WORD   T20
002174  033330                                                 .WORD   T21
002176  033646                                                 .WORD   T22
002200  034222                                                 .WORD   T23
002202  035360                                                 .WORD   T24
002204  036322                                                 .WORD   T25
002206  036616                                                 .WORD   T26
1157
```

```
1172
1173
1174                                 .SBTTL  DEFAULT HARDWARE P-TABLE
1175
1176                                 ;++
1177                                 ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1178                                 ; THE TEST DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
1179                                 ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
1180                                 ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
1181                                 ;--
1182
1183 002210                                  BGNHW   DFPTBL
     002210  000004                                                                   .WORD   L10000-L$HW/2
     002212                                                                   L$HW::
     002212                                                                   DFPTBL::
1184
1185 002212  160460                          .WORD   160460  ;DEFAULT CSR ADDRESS
1186 002214  000310                          .WORD   310     ;DEFAULT VECTOR ADDRESS
1187 002216  177777                          .WORD   177777  ;DEFAULT ACTIVE LINES BIT MAP
1188 002220     005                          .BYTE   5       ;DEFAULT BR LEVEL
1189                                         .EVEN
1190
1191 002222                                  ENDHW
     002222                                                                   L10000:
```

```
                                .SBTTL  SOFTWARE P-TABLE

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

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

002224  000001                        OPTION::        .WORD   21      ;BIT MAP OF PROGRAM CONTROL FLAGS
002226  000000                        NDERPT::        .WORD   0       ;DEFAULT NUMBER OF INDIVIDUAL DATA ERRORS TO RPT.

                                        ENDSW
002230                                                                  L10001:
```

```
                        .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

002230                          EQUALS

                        ; BIT DIFINITIONS

100000                  BIT15== 100000
040000                  BIT14== 40000
020000                  BIT13== 20000
010000                  BIT12== 10000
004000                  BIT11== 4000
002000                  BIT10== 2000
001000                  BIT09== 1000
000400                  BIT08== 400
000200                  BIT07== 200
000100                  BIT06== 100
000040                  BIT05== 40
000020                  BIT04== 20
000010                  BIT03== 10
000004                  BIT02== 4
000002                  BIT01== 2
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                  PRI7== 340
000300                  PRI6== 300
000240                  PRI5== 240
000200                  PRI4== 200
000140                  PRI3== 140
000100                  PRI2== 100
000040                  PRI1== 40
000000                  PRI0== 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
1268
```

```
1277
1278
1279
1280                                    .SBTTL  GLOBAL DATA SECTION
1281
1282                                    ;++
1283                                    ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1284                                    ; IN MORE THAN ONE TEST.
1285                                    ;--
1286
1287
1288                                    ;*****************************************************************
1289                                    ;       UNIT VARIBLE AREA
1290                                    ;*****************************************************************
1291
1292 002230  177777                     ACTLNS:: .WORD  177777  ;ACTIVE LINE BIT MAP.
1293 002232  000300                     RXVECA:: .WORD  300     ;RX VECTOR ADDRESS.
1294 002234  000304                     TXVECA:: .WORD  304     ;TX VECTOR ADDRESS.
1295 002236  000000                     UNITN:: .WORD   0       ;UNIT NUMBER.
1296 002240     004                     BRLEVL:: .BYTE  4       ;INTERRUPT BUS REQUEST LEVEL
1297                                             .EVEN
1298
1299                                    ;*****************************************************************
1300                                    ;       DEVICE REGISTER ADDRESS TABLE
1301                                    ;*****************************************************************
1302 002242                             DRADRT::
1303 002242  160020                     CSRA::   .WORD  160020  ;DHU-11 CSR ADDRESS.
1304 002244  160022             RXTMA:: RBUFA:: .WORD   160022  ;DHU-11 RECIEVE BUFFER/TIMER ADDRESS.
1305 002246  160024                     LPRA::   .WORD  160024  ;DHU-11 LINE PARAMETER REGISTER ADDRESS.
1306 002250  160026             FDATA:: FSLSA:: .WORD   160026  ;DHU-11 FIFO SIZE/LINE STATUS REGISTER ADDRESS,
1307                                             ;AND FIFO DATA REGISTER ADDRESS.
1308 002252  160030                     LNCTRA:: .WORD  160030  ;DHU-11 LINE CONTROL REGISTER ADDRESS.
1309 002254  160032                     TXAD1A:: .WORD  160032  ;DHU-11 TRANSMIT BUFFER 1 REGISTER ADDRESS
1310 002256  160034                     TXAD2A:: .WORD  160034  ;DHU-11 TRANSMIT BUFFER 2 REGISTER ADDRESS
1311 002260  160036                     TXBFCA:: .WORD  160036  ;DHU-11 TRANSMIT BUFFER COUNT REGISTER ADDRESS
1312
1313                                    ;*****************************************************************
1314                                    ;       REGISTER MESSAGE ADDRESS TABLE
1315                                    ;*****************************************************************
1316 002262  005464                     RMATBB:: .WORD  DR00MG  ;ADDRESS OF "CSR" MESSAGE.
1317 002264  005470                              .WORD  DR02MG  ;ADDRESS OF "RBUF" MESSAGE.
1318 002266  005475                              .WORD  DR04MG  ;ADDRESS OF "LPR" MESSAGE.
1319 002270  005501                              .WORD  DR06MG  ;ADDRESS OF "STAT" MESSAGE.
1320 002272  005517                              .WORD  DR10MG  ;ADDRESS OF "LNCTRL" MESSAGE.
1321 002274  005526                              .WORD  DR12MG  ;ADDRESS OF "TBUFFAD1" MESSAGE.
1322 002276  005537                              .WORD  DR14MG  ;ADDRESS OF "TBUFAD2" MESSAGE.
1323 002300  005550                              .WORD  DR16MG  ;ADDRESS OF "TBUFFCT" MESSAGE.
1324
1325                                    ;*****************************************************************
1326                                    ;       ASSORTED GLOBAL VARIABLES:
1327                                    ;*****************************************************************
1328 002302  000000                     BUFPTR:: .WORD  0       ;STORAGE FOR RECEIVE CHARACTER BUFFER POINTER.
1329 002304  000000                     EXOERR:: .WORD  0       ;"EXIT ON ERROR" FLAG.
1330 002306  000000                     CTRLCF:: .WORD  0       ;STORAGE FOR THE CONTROL-C FLAG.
1331 002310  000000                     IESTAT:: .WORD  0       ;STORAGE FOR THE INTERRUPT ENABLE BIT STATES.
1332 002312  000000                     PASCNT:: .WORD  0       ;STORAGE FOR PASS COUNT USED IN ROM VERSION TEST.
1333 002314  000000                     RXINTC:: .WORD  0       ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
```

GLOBAL DATA SECTION

```
1334 002316  000000                    RXINTF:: .WORD  0       ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
1335 002320  000000                    TP4FLG:: .WORD  0       ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
1336 002322  000000                    TP4VEC:: .WORD  0       ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
1337 002324  000001                    TSTNUM:: .WORD  1       ;STORAGE FOR THE TEST NUMBER.
1338 002326  000000                    TXINTC:: .WORD  0       ;STORAGE FOR TRANSMIT INTERRUPT COUNT.
1339 002330  000000                    TXINTF:: .WORD  0       ;STORAGE FOR TRANSMIT INTERRUPT FLAGS.
1340 002332  000000                    WORD1::  .WORD  0       ;LOCATION FOR PASSING INDIRECT PARAMETERS.
1341
1342                            ;*******************************************************************************
1343                            ;       LINE TIME CLOCK VARIABLES AND STORAGE.
1344                            ;*******************************************************************************
1345 002334  177546                    CLKCSR:: .WORD  177546  ;CSR ADDRESS OF THE LTC.
1346 002336  000300                    CLKBRL:: .WORD  PRIO6   ;INTERRUPT PRIORITY LEVEL OF THE LTC.
1347 002340  000100                    CLKVEC:: .WORD  100     ;INTERRUPT VECTOR ADDRESS OF THE LTC.
1348 002342  000074                    CLKHRZ:: .WORD  60.     ;INTERRUPT FREQUENCY OF THE LTC.
1349 002344  000000                    TIMER1:: .WORD  0       ;HARDWARE CLOCK COUNTER #1.
1350 002346  000000                    TIMER2:: .WORD  0       ;HARDWARE CLOCK COUNTER #2.
1351 002350  000170                    TIMER3:: .WORD  120.    ;HARDWARE BREAK COUNTER LOCATION.
1352 002352  000170                    BCOUN1:: .WORD  120.    ;BREAK COUNT VALUE IN CLOCK TICKS.
1353 002354  000021                    MSTICK:: .WORD  17.     ;NUMBER OF MILLI-SECONDS PER LTC TICK.
1354 002356  000062                    MSLCNT:: .WORD  62      ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.
1355
1356                            ;*******************************************************************************
1357                            ;       MEMMORY MANAGEMENT VARIABLES AND FLAGS.
1358                            ;*******************************************************************************
1359 002360  177572                    MMSR0:: .WORD   177572  ;ADDRESS OF MEM MGT STATUS REGISTER #0.
1360 002362  000000                    MMPRES:: .WORD  0       ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
1361 002364  000000                    MMENAB:: .WORD  0       ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).
1362 002366  172340                    PAR0A:: .WORD   172340  ;ADDRESS OF MEM MGT PAR #0.
1363
1364                            ;*******************************************************************************
1365                            ;       BIT MASK TABLE OF UN-USED DHU DEVICE REGISTER BITS.
1366                            ;*******************************************************************************
1367 002370  137660                    UNBTTB::.WORD   137660  ;UNUSED BIT MASK FOR THE CSR
1368 002372  177777                            .WORD   177777  ;UNUSED BIT MASK FOR THE RBUF/RXTIMER REG
1369 002374  000007                            .WORD   7       ;UNUSED BIT MASK FOR THE LPR
1370 002376  177777                            .WORD   177777  ;UNUSED BIT MASK FOR THE STAT/FIFOSIZE/DATA REG
1371 002400  166051                            .WORD   166051  ;UNUSED BIT MASK FOR THE LNCTRL
1372 002402  000000                            .WORD   0       ;UNUSED BIT MASK FOR THE TBUFFAD1
1373 002404  177774                            .WORD   177774  ;UNUSED BIT MASK FOR THE TBUFFAD2
1374 002406  000000                            .WORD   0       ;UNUSED BIT MASK FOR THE TBUFFCT
1375
1376                            ;*******************************************************************************
1377                            ;       TABLE OF WORDS WITH CORRESPONDING BIT SET FOR GENERATION OF BIT MAPS.
1378                            ;*******************************************************************************
1379 002410  000001    BITTBL:: .WORD  1               ;BIT 0 SET.
1380 002412  000002             .WORD  2               ;BIT 1 SET.
1381 002414  000004             .WORD  4               ;BIT 2 SET.
1382 002416  000010             .WORD  10              ;BIT 3 SET.
1383 002420  000020             .WORD  20              ;BIT 4 SET.
1384 002422  000040             .WORD  40              ;BIT 5 SET.
1385 002424  000100             .WORD  100             ;BIT 6 SET.
1386 002426  000200             .WORD  200             ;BIT 7 SET.
1387 002430  000400             .WORD  400             ;BIT 8 SET.
1388 002432  001000             .WORD  1000            ;BIT 9 SET.
1389 002434  002000             .WORD  2000            ;BIT 10 SET.
1390 002436  004000             .WORD  4000            ;BIT 11 SET.
```

GLOBAL DATA SECTION

```
1391 002440  010000                    .WORD   10000           ;BIT 12 SET.
1392 002442  020000                    .WORD   20000           ;BIT 13 SET.
1393 002444  040000                    .WORD   40000           ;BIT 14 SET.
1394 002446  100000                    .WORD   100000          ;BIT 15 SET.
1395
1396                            ;*******************************************************************************
1397                            ;*      GPR SAVE AREA ZERO.
1398                            ;*******************************************************************************
1399 002450                    GPRSOB::                        ;BASE OF GPR SAVE AREA NUMBER ZERO.
1400 002450  000000                    .WORD   0               ;WORD 1, STORAGE FOR R1.
1401 002452  000000                    .WORD   0               ;WORD 2, STORAGE FOR R2.
1402 002454  000000                    .WORD   0               ;WORD 3, STORAGE FOR R3.
1403 002456  000000                    .WORD   0               ;WORD 4, STORAGE FOR R4.
1404 002460  000000                    .WORD   0               ;WORD 5, STORAGE FOR R5.
1405
1406                            ;*******************************************************************************
1407                            ;*      TRANSMISSION AND RECEPTION VARIABLES, POINTERS, AND FLAGS.
1408                            ;*******************************************************************************
1409 002462  000000            ERSMRP:: .WORD  0               ;ERROR SUMMARY REPORT FLAGS.
1410 002464                    ERCNTB:: .BLKW  16.             ;TABLE OF ERROR COUNTERS.
1411
1412                            ;*******************************************************************************
1413                            ;       STORAGE AREA FOR THE BMP CODE QUEUE.
1414                            ;*******************************************************************************
1415 002524  000000            BMPCQP:: .WORD  0               ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
1416 002526                    BMPCQB:: .BLKW  64.             ;STORAGE FOR 32 CELLS, TEST0 PLUS BMP CODE.
1417 002726                    BMPCQE::                        ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1418
1419                            ;*******************************************************************************
1420                            ;       GENERAL TABLE AND BUFFER AREA--513 WORDS.
1421                            ;*******************************************************************************
1422 002726                    BUFBAS::                        ;BASE OF MEMORY BUFFER.
1423 002726                    ERLTBL::        .BLKW   128.    ;FIRST HALF OF GENERAL TABLE OR BUFFER.
1424 003326                    BUFMID::        .BLKW   64.     ;SECOND HALF OF GENERAL TABLE OR BUFFER.
1425 003526                    BUF3QT::        .BLKW   64.     ;LAST QUARTER OF THE BUFFER AREA.
1426 003726                    BUFEND::                        ;END OF GENERAL PURPOSE MEMORY BUFFER.
1427 003726                    ENDETB::        .BLKW   16.     ;BUFFER OVERFLOW SPACE.
1428
1429
1430
1443 003766                            ERRTBL
     003766                                                                    L$ERRTBL::
     003766  000000            ERRTYP::        .WORD   0
     003770  000000            ERRNBR::        .WORD   0
     003772  000000            ERRMSG::        .WORD   0
     003774  000000            ERRBLK::        .WORD   0
1444
1445                                    .EVEN
```

```
1447                    .SBTTL  GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.
1448                    ;*************************************************************************
1449                    ;*      THERE ARE 4 ROUTINES AND MACRO DEFINITIONS USED FOR THE HANDLING OF
1450                    ;*      GPR VALUES DURING SUBROUTINE CALLS WITHIN THIS PROGRAM.  THE FOUR
1451                    ;*      ROUTINES/MACRO CALLS HAVE THE FOLLOWING NAMES:
1452                    ;*
1453                    ;*      SAVE - MACRO DEFINITION USED AT THE BEGINNING OF A SUBROUTINE TO
1454                    ;*             SAVE THE GPR CONTENTS FOR LATER RESTORATION.
1455                    ;*      PASS - MACRO DEFINITION USED AT THE END OF A SUBROUTINE TO RESTORE
1456                    ;*             THE PREVIOUSLY SAVED GPR CONTENTS AND TO LEAVE THE CONTENTS
1457                    ;*             OF THE SPECIFIED GPR(S) INTACT (NOT RESTORED).
1458                    ;*      PREG05 - SUBROUTINE WHICH IS CALLED FROM THE SAVE AND PASS MACRO
1459                    ;*             EXPANSIONS WHICH ACTUALLY PERFORMS THE ACTIONS ON THE GPRS.
1460                    ;*
1461                    ;*      DURING A SUBPOUTINE WHICH USES THESE GPR SAVE ROUTINES THE VALUES
1462                    ;*      OF THE GPRS ARE STORED ON THE STACK IN THE FOLLOWING STACK FRAME:
1463                    ;*
1464                    ;*             SP     -> RET PC INTO PREG05 ROUTINE.
1465                    ;*             SP+2   -> GPR R0 CONTENTS.
1466                    ;*             SP+4   -> GPR R1 CONTENTS.
1467                    ;*             SP+6   -> GPR R2 CONTENTS.
1468                    ;*             SP+8   -> GPR R3 CONTENTS.
1469                    ;*             SP+10  -> GPR R4 CONTENTS.
1470                    ;*             SP+12  -> GPR R5 CONTENTS.
1471                    ;*             SP+14  -> RET PC INTO CALLER OF SUBROUTINE WHICH CALLED PREG05.
1472                    ;*
1473                    ;*      EACH LEVEL OF SUBROUTINE CALLING USES 8 WORDS OF STACK OVERHEAD.
1474                    ;*      THE SAVE AND PASS MACROS CAN ALSO BE USED IN "STRAIGHT LINE CODE"
1475                    ;*      TO SAVE AND RESTORE THE GPR VALUES.  IN ANY CASE, AFTER THE
1476                    ;*      ISSUING OF A PASS CALL THE GPRS WILL BE RESTORED TO THE VALUES
1477                    ;*      THEY HAD PRIOR TO THE LAST SAVE CALL (EXCEPT FOR THE EXCEPTED,
1478                    ;*      OR PASSED INTACT, GPRS SPECIFIED AS PARAMETERS TO THE PASS CALL)
1479                    ;*      AND THE SP WILL ALSO BE RESTORED TO  TS CONDITION BEFORE THE LAST
1480                    ;*      SAVE CALL.  THE PROGRAMMER MUST BE SURE THAT THE SP HAS THE SAME
1481                    ;*      VALUE WHEN THE PASS MACRO IS CALLED AS IT HAD IMMEDIATELY AFTER
1482                    ;*      THE SAVE MACRO WAS CALLED.
1483                    ;*************************************************************************
```

GPR FRAME ACCESS EQUATES

```
1485                                    .SBTTL GPR FRAME ACCESS EQUATES
1486                                            ;+++
1487                                            ;EQUATES THAT ALLOW ACCESS TO THE STACK FRAME.  THESE ARE THE
1488                                            ;OFFSETS INTO THE STACK FOR REGISTERS SAVED DURING THE PREGO5
1489                                            ;ROUTINE.
1490                                            ;---
1491
1492            000036                          LPCSLT==        36      ;OFFSET FOR LAST RETURN PC.
1493            000016                          PCSLOT==        16      ;OFFSET FOR RETURN PC.
1494            000014                          R5SLOT==        14      ;OFFSET FOR R5.
1495            000012                          R4SLOT==        12      ;OFFSET FOR R4.
1496            000010                          R3SLOT==        10      ;OFFSET FOR R3.
1497            000006                          R2SLOT==        6       ;OFFSET FOR R2.
1498            000004                          R1SLOT==        4       ;OFFSET FOR R1.
1499            000002                          ROSLOT==        2       ;OFFSET FOR R0.
```

```
1501                    .SBTTL  GLOBAL MACRO DEFINITION         - SAVE -
1502                    ;*************************************************************************
1503                    ;*      THIS MACRO IS USED AT THE BEGINNING OF A SUBROUTINE TO SAVE THE
1504                    ;*      CONTENTS OF THE GPRS R0 THRU R5.
1505                    ;*
1506                    ;* INPUTS:      SP - UNCHANGED SINCE SUBROUTINE WAS ENTERED
1507                    ;*              R5SLOT - OFFSET TO STACK SLOT FOR R5 (EQUATED TO 14 OCTAL)
1508                    ;*
1509                    ;* OUTPUTS:     GPR SAVE AREA ON THE STACK IS LOADED WITH THE CONTENTS OF GPRS
1510                    ;*              TOP OF STACK - LOADED WITH THE RETURN ADDRESS INTO PREG05
1511                    ;*
1512                    ;* CALLING SEQUENCE:    SAVE
1513                    ;*
1514                    ;* COMMENTS:    NO ARGUMENTS ARE ALLOWED.
1515                    ;*              THE PASS MACRO SHOULD BE CALLED TO RESTORE THE GPR VALUES.
1516                    ;*
1517                    ;* SUBORDINATE ROUTINES CALLED: PREG05.
1518                    ;*************************************************************************
1519
1520                            .MACRO  SAVE
1521                            .LIST
1522                                    JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
1523                            .NLIST
1524                            .ENDM   SAVE
```

```
1526                    .SBTTL  GLOBAL MACRO DEFINITION          - PASS -
1527                    ;*********************************************************************
1528                    ;*      THIS MACRO IS USED IN CONJUNCTION WITH THE SAVE MACRO.  IT IS
1529                    ;*      CALLED AT END OF A SUBROUTINE TO PASS PARAMETERS IN GPRS BACK TO THE
1530                    ;*      CALLING ROUTINE BY ALTERING THE GPR SAVE AREA ON THE STACK AND THEN
1531                    ;*      RETURNING TO PREG05 TO RESTORE THE GPRS TO THEIR SAVED VALUES.
1532                    ;*
1533                    ;* INPUTS:      ONLY ALLOWED ARGUMENTS ARE "R0" THRU "R5".
1534                    ;*              R0SLOT THRU R5SLOT MUST BE EQUATED TO THEIR RESPECTIVE GPR SAVE
1535                    ;*                      SLOT OFFSETS BEFORE CALLING THIS MACRO.
1536                    ;*
1537                    ;* OUTPUTS:     THE GPR VALUES ARE PUT IN THEIR RESPECTIVE SLOTS ON THE STACK.
1538                    ;*
1539                    ;* CALLING SEQUENCE:    PASS    R0,R1,...
1540                    ;*
1541                    ;* COMMENTS:    ANY COMBINATION OF GPR ARGUMENTS MAY BE LISTED IN ANY ORDER.
1542                    ;*              FOR EXAMPLE, THE FOLLOWING ARE LEGAL:
1543                    ;*                      PASS    R1
1544                    ;*                      PASS    R4,R0,R2
1545                    ;*              THE GPRS LISTED AS ARGUMENTS WILL BE PASSED INTACT TO THE
1546                    ;*              CALLING ROUTINE, ALL OTHER GPRS WILL BE RESTORED.
1547                    ;*              THE SP MUST BE AT ITS ORIGINAL VALUE WHEN PASS IS CALLED.
1548                    ;*
1549                    ;*              THE MACRO CALL
1550                    ;*                      PASS    R0,R3
1551                    ;*              EXPANDS INTO THE FOLLOWING ASSEMBLY CODE:
1552                    ;*                      MOV     R0,R0SLOT(SP)           ;PUT R0 IN STACK SLOT.
1553                    ;*                      MOV     R3,R3SLOT(SP)           ;PUT R3 IN STACK SLOT.
1554                    ;*                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
1555                    ;*              IN THIS EXAMPLE GPRS R1, R2, R4, AND R5 WILL BE RESTORED TO
1556                    ;*              THEIR VALUES CONTAINED IN THE STACK FRAME AND R0 AND R3
1557                    ;*              WILL BE LEFT AT THEIR VALUES PRIOR TO THIS PASS CALL.
1558                    ;*
1559                    ;* SUBORDINATE ROUTINES CALLED: (PREGRT - LABEL WITHIN PREG05, VALUE ON STACK.)
1560                    ;*********************************************************************
1561
1562                            .MACRO  PASS    A,B,C,D,E,F
1563                            .IRP    X,<A,B,C,D,E,F>
1564                            .IF     NB,X
1565                            .LIST
1566                                    MOV     X,X'SLOT(SP)            ;PUT X IN STACK SLOT.
1567                            .NLIST
1568                            .ENDC
1569                            .ENDM
1570                            .LIST
1571                                    JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
1572                            .NLIST
1573                            .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.
                        ;*************************************************************************

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

004016  004736                  JSR     PC,@(SP)+       ;CALL THE SUBROUTINE AT THE RETURN ADDRESS
                                                        ;FROM THE PREG05 CALL, PUTTING THE PRESENT
                                                        ;PC ON THE STACK AS A RETURN ADDRESS INTO
                                                        ;THIS (PREG05) ROUTINE.

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

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

004034  000205                  RTS     R5              ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05.
                                                          ;RESTORING R5 IN THE PROCESS.
```

GLOBAL TEXT SECTION

```
1629                                    .SBTTL  GLOBAL TEXT SECTION
1637
1638
1639                                    ;++
1640                                    ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1641                                    ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1642                                    ; MORE THAN ONE TEST.
1643                                    ;
1644
1645                                    ;
1646                                    ; NAMES OF DEVICES SUPPORTED BY PROGRAM
1647                                    ;
1648 004036                                     DEVTYP  <DHU-11>
     004036                                                                    L$DVTYP::
     004036  104     110     125                                                    .ASCIZ  /DHU-11/
     004041  055     061     061
     004044  000
                                                                                    .EVEN
1649
1655
1656                                    ; TEST DESCRIPTION
1657                                    ;
1658 004046                                     DESCRIPT        <DHU-11 FUNC TST PART1>
     004046                                                                    L$DESC::
     004046  104     110     125                                                    .ASCIZ  /DHU-11 FUNC TST PAR
T1/
     004051  055     061     061
     004054  040     106     125
     004057  116     103     040
     004062  124     123     124
     004065  040     120     101
     004070  122     124     061
     004073  000
                                                                                    .EVEN
1659                                            .EVEN
1666
1667
```

```
1676
1677                    .NLIST BIN
1678
1679                            ; ********* FORMAT STATEMENTS USED IN PRINT CALLS *********
1680
1681 004074  EF0503:: .ASCIZ /%T%N/
1682 004101  EF0505:: .ASCIZ /%A    %D5%A ILLEGAL INTERRUPTS RECEIVED.%N/
1683 004154  EF1401:: .ASCIZ /%N%A ROM VERSION NUMBERS: PROC 1 = %D2%A(D)   PROC 2 = %D2%A(D)%N/
1684 004256  EF1402:: .ASCIZ /%T%A ROM VERSION NUMBER %T%N/
1685 004313  EF1601:: .ASCIZ /%A  %T%A, TEST ABORTED %N/
1686 004345  EF1602:: .ASCIZ /%A    EXPECTED DATA: %O6%A (O).%N/
1687 004407  EF1603:: .ASCIZ /%A    ACTUAL DATA:   %O6%A (O).%N/
1688 004451  EF1604:: .ASCIZ /%A  BAD BIT(S) IN DEVICE %T%A REGISTER FOR LINE %D2%A (D).%N/
1689 004546  EF3001:: .ASCIZ /%A    EXPECTED OR CORRECT VALUE: %O3%N/
1690 004615  EF3002:: .ASCIZ /%A    ACTUAL OR MEASURED  VALUE: %O3%N/
1691 004664  EF9006:: .ASCIZ /%A  %T%A %D2%A(D)%N/
1692 004710  EF9010:: .ASCIZ /%A    NUMBER OF ERRORS DETECTED ON LINE %D2%A IS %D5%N/
1693 004777  EF9016:: .ASCIZ /%A  UNEXPECTED %T%A FOR LINE %D2%A(D) IN FIFO AFTER RESET:%N/
1694 005074  EF9017:: .ASCIZ /%A    %T%A (WITH ERROR FLAGS) IS %O6%A(O)%N/
1695 005150  EF9018:: .ASCII /%A  %T%A IN SELFTEST CODE FIFO SLOT FOR LINE %D2/
1696 005230           .ASCIZ /%A(D) AFTER RESET.%N/
1697 005255  EF9301:: .ASCIZ /%A %T%D2%A(D),  BMP CODE REPORTED :%O3%A(O)%N/
1698 005333  EF9302:: .ASCIZ /%A  OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)%N/
1699 005433  MFUNIT:: .ASCIZ /%N%A TESTING UNIT :%D4%N/
1700                  .EVEN
1701                                  .LIST BIN
```

```
1710
1711                    .NLIST BIN
1712
1713
1714                            ;********* GLOBAL ERROR MESSAGES *************
1715
1716 005464  DR00MG:: .ASCIZ /CSR/
1717 005470  DR02MG:: .ASCIZ /RBUF/
1718 005475  DR04MG:: .ASCIZ /LPR/
1719 005501  DR06MG:: .ASCIZ /FIFOSIZE,STAT/
1720 005517  DR10MG:: .ASCIZ /LNCTRL/
1721 005526  DR12MG:: .ASCIZ /TBUFFAD1/
1722 005537  DR14MG:: .ASCIZ /TBUFFAD2/
1723 005550  DR16MG:: .ASCIZ /TBUFFCT/
1724 005560  EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
1725 005616  EM0201:: .ASCIZ / MASTER RESET TEST FAILED./
1726 005651  EM0202:: .ASCIZ /  MASTER RESET BIT DID NOT CLEAR AFTER BOARD RESET./
1727 005735              .ASCIZ /    WAITED 5 SECONDS.  BIT DEFECTIVE OR FIRMWARE HUNG./
1728 006024  EM0203:: .ASCIZ /  MASTER RESET BIT CLEAR IMMEDIATELY AFTER BOARD RESET./
1729 006114              .ASCIZ /    BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
1730 006167  EM0204:: .ASCIZ \  MR BIT WENT CLEAR WITHIN 1/2 SECOND OF BOARD RESET.\
1731 006255              .ASCIZ /    BIT DEFECTIVE OR SELFTEST WAS (INCORRECTLY) SKIPPED./
1732 006346  EM0301:: .ASCIZ /MASTER RESET (SKIP SELFTEST) TEST FAILED./
1733 006420  EM0302:: .ASCIZ /  MR BIT CLR WITHIN 10 MILISECOND AFTER BOARD RESET./
1734 006505              .ASCIZ /    BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
1735 006560  EM0303:: .ASCIZ \  MR BIT WENT CLEAR 1/5 TO 5 SECONDS AFTER RESET.\
1736 006662              .ASCIZ /    SELFTEST DID NOT GET SKIPPED (SHOULD HAVE BEEN SKIPPED)./
1737 006737  EM0401:: .ASCIZ /RBUF REGISTER RX CHARACTER FIELD TEST FAILED./
1738 007015  EM0402:: .ASCIZ /  IMPROPER CODE FOUND IN RX FIFO AFTER DUT RESET./
1739 007077              .ASCIZ /    EXPECTED: SELFTEST CODE,   ACTUAL: IMPROPER CODE./
1740 007165  EM0501:: .ASCIZ /RBUF REGISTER ERROR FLAGS FIELD TEST FAILED/
1741 007241  EM0502:: .ASCIZ /  RX ERROR FLAG(S) FOUND CLEAR ON SELFTEST CODE./
1742 007322              .ASCIZ /    EXPECTED: ALL ERROR FLAGS SET,  ACTUAL: FLAG(S) CLEAR./
1743 007415  EM0525:: .ASCIZ /  RX INTERRUPT(S) RECEIVED WITH RX INTERRUPTS DISABLED./
1744 007505  EM0526:: .ASCIZ /  TX INTERRUPT(S) RECEIVED WITH TX INTERRUPTS DISABLED./
1745 007575  EM0601:: .ASCIZ /CSR RX.DATA.AVAIL BIT TEST FAILED/
1746 007637  EM0602:: .ASCIZ /  RX.DATA.AVAIL BIT FOUND CLEAR AFTER RESET COMPLETION./
1747 007727              .ASCIZ /    EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
1748 010017  EM0603:: .ASCIZ /  RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO./
1749 010111              .ASCIZ /    600 CHARS READ FROM FIFO WITHOUT R.D.A BIT CLEARING./
1750 010202  EM0701:: .ASCIZ /RBUF RX.DATA.VALID BIT TEST FAILED/
1751 010245  EM0702:: .ASCIZ /  RX.DATA.VALID BIT FOUND CLEAR AFTER RESET COMPLETION./
1752 010335              .ASCIZ /    EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
1753 010425  EM0703:: .ASCIZ /  RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO./
1754 010517              .ASCIZ /    600 CHARS READ FROM FIFO WITHOUT R.D.V BIT CLEARING./
1755 010610  EM0801:: .ASCIZ /RBUF RX.LINE.NUMBER FIELD TEST FAILED/
1756 010656  EM0802:: .ASCIZ /  LINE NUMBER WRONG ON A SELFTEST CODE./
1757 010726  EM0901:: .ASCIZ /CHECK FOR BMP_CODES TEST FAILED/
1758 010766  EM0902:: .ASCIZ /UNEXPECTED BMP CODES FOUND./
1759 011022  EM1001:: .ASCIZ /SKIP SELF-TEST TEST FAILED/
1760 011055  EM1002:: .ASCIZ /  SKIP SELF-TEST TOOK TOO LONG TO COMPLETE, > 50 MS./
1761 011142  EM1003:: .ASCIZ /  SKIP SELF-TEST COMPLETED TOO SOON, < 10 MS./
1762 011220  EM1101:: .ASCIZ /DIAGNOSTIC FAIL (SKP SELFTEST) TEST FAILED/
1763 011273  EM1201:: .ASCIZ /SELF-TEST TEST FAILED/
1764 011321  EM1202:: .ASCIZ /  SELF-TEST TOOK TOO LONG TO COMPLETE, > 5 SECONDS./
1765 011405  EM1203:: .ASCIZ \  SELF-TEST COMPLETED TOO SOON, < 1/2 SECOND.\
1766 011463  EM1204:: .ASCIZ /  SELF-TEST DID NOT EXECUTE/
```

```
1767 011517  EM1205:: .ASCIZ /  DIAG_FAIL BIT BAD/
1768 011543  EM1301:: .ASCIZ /FAIL SELF-TEST TEST FAILED/
1769 011576  EM1302:: .ASCIZ / SELF-TEST ERROR REPORTING BAD/
1770 011635  EM1401:: .ASCIZ /ROM VERSION_NUMBER TEST FAILED/
1771 011674  EM1402:: .ASCIZ /  FIFO EMPTY, ONE OR MORE ROM VERSION_NUMBERS MISSING/
1772 011762  EM1403:: .ASCIZ /  ROM VERSION_NUMBER FOUND OUT OF SEQUENCE/
1773 012035  EM1404:: .ASCIZ /  ONE OR MORE ROM VERSION_NUMBERS MISSING/
1774 012107  EM1405:: .ASCIZ /    PROC_1/
1775 012122  EM1406:: .ASCIZ /    PROC_2/
1776 012135  EM1407:: .ASCIZ /NOT FOUND/
1777 012147  EM1408:: .ASCIZ /FOUND/
1778 012155  EM1501:: .ASCIZ /CSR BIT 4 TEST FAILED/
1779 012203  EM1502:: .ASCIZ /CSR BIT 4 BAD/
1780 012221  EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
1781 012304  EM1604:: .ASCIZ \DEVICE REGISTER WORD READ/WRITE TEST FAILED\
1782 012360  EM1701:: .ASCIZ \DEVICE REGISTER WORD READ/MODIFY/WRITE TEST FAILED\
1783 012443  EM1801:: .ASCIZ \DEVICE REGISTER BYTE READ/WRITE TEST FAILED\
1784 012517  EM1901:: .ASCIZ \DEVICE REGISTER BYTE READ/MODIFY/WRITE TEST FAILED\
1785 012602  EM2001:: .ASCIZ /DEVICE STAT REGISTER ID BIT TEST FAILED/
1786 012652  EM2002:: .ASCIZ /ID BIT BAD.  EXPECTED: SET, ACTUAL: CLEAR./
1787 012725  EM2301:: .ASCIZ /TX_ENABLE (INACTIVE) BIT TEST FAILED/
1788 012772  EM2302:: .ASCIZ /  TX_ENABLE BIT BAD ON LINE: /
1789 013030  EM2401:: .ASCIZ /TX_ENABLE (ACTIVE) BIT TEST FAILED/
1790 013073  EM2601:: .ASCIZ /RECEIVE INTERRUPT TEST FAILED/
1791 013131  EM2602:: .ASCIZ /  NO RX INT GENERATED (DATA_VALID SET, RX INTS ENABLED)./
1792 013222  EM2603:: .ASCIZ /  NO RX INT GENERATED (NO CODES IN FIFO AFTER RESET)./
1793 013310  EM2604:: .ASCIZ /  NO RX INT GENERATED (RX_DATA_AVAIL CLR, RX INTS ENABLED)./
1794 013404  EM2605:: .ASCIZ /  RX INTERRUPT GENERATED WITH RX_DATA_AVAIL CLEAR./
1795 013467  EM2606:: .ASCIZ /TRANSMIT INTERRUPT TEST ERROR:/
1796 013526  EM2607:: .ASCIZ /  TX ACTION SET REPEATEDLY AFTER BOARD RESET, NO DATA SENT./
1797 013622  EM2608:: .ASCIZ /  TX_ACTION STUCK SET AFTER BOARD RESET./
1798 013673  EM2609:: .ASCIZ /  TX INTERRUPT GENERATED WITH TX_ACTION CLEAR./
1799 013752  EM2610:: .ASCIZ /  NO TX INTERRUPT WITH TX_ACTION SET AND TX INTS ENABLED./
1800 014044  EM2611:: .ASCIZ /  TX_ACTION NOT SET AFTER CHARS SENT ON ALL LINES./
1801 014127  EM2612:: .ASCIZ /  NO RX INT GENERATED (RX_DATA_AVAIL SET, RX INTS ENABLED)./
1802 014223  EM3001:. .ASCIZ /INTERRUPT BR LEVEL TEST FAILED/
1803 014262  EM3002:: .ASCIZ /  NO RX_DATA_AVAIL FROM SELFTEST CODES IN FIFO AFTER RESET./
1804 014356  EM3003:: .ASCIZ /  TX INTERRUPT GENERATED AT WRONG BR LEVEL:/
1805 014432  EM3004:: .ASCIZ /  RX INTERRUPT GENERATED AT WRONG BR LEVEL:/
1806 014506  EM3005:: .ASCIZ /  TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS RX INT./
1807 014600  EM3101:: .ASCIZ /DIAGNOSTIC FIELD (BMP) TEST FAILED/
1808 014643  EM3102:: .ASCIZ /  DIAGNOSTIC FIELD (BMP REQUEST) BAD ON LINE: /
1809 014722  EM9014:: .ASCIZ /SUMMARY REPORTS FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:/
1810 015016  EM9017:: .ASCII /  FIFO WILL NOT PURGE (DATA.VALID STUCK SET),/
1811 015073          .ASCIZ / REMAINDER OF TEST SKIPPED./
1812 015127  EM9018:: .ASCIZ /NO CODE/
1813 015137  EM9019:: .ASCIZ /NON-SELFTEST/
1814 015154  EM9020:: .ASCIZ /SELFTEST ERROR CODE/
1815 015200  EM9022:: .ASCIZ /DATA CHARACTER/
1816 015217  EM9023:: .ASCIZ /MODEM STATUS CODE/
1817 015241  EM9024:: .ASCIZ /SELFTEST CODE/
1818 015257  EM9301:: .ASCIZ /BMP CODES WERE REPORTED DURING THIS DIAGNOSTIC/
1819 015336  EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
1820 015366  EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
1821 015433  EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
1822                  .EVEN
1823                          .LIST BIN
```

```
1832
1833
1834                            .SBTTL  GLOBAL ERROR REPORT SECTION
1835
1836                            ;++
1837                            ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
1838                            ; USED BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
1839                            ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
1840                            ;--
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0101 -
                        ;*********************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
                        ;*      INFORMATION IF AN ERROR IS DETECTED IN TEST 1 (REGISTER ADDRESS
                        ;*      ACCESS TEST). IF THE "EXTENDED ERROR INFO" OPTION HAS BEEN SELECTED
                        ;*      THEN THIS SUBROUTINE WILL REPORT THE TYPE OF ACCESS (READ OR WRITE OR
                        ;*      BOTH) WHICH CAUSED A BUS TIME-OUT TRAP (004 TRAP).A MESSAGE INDICATING
                        ;*      THAT THE DHU MAY BE AT THE WRONG UNIBUS ADDRESS IS ALSO PRINTED.
                        ;*
                        ;* INPUTS:      R5 - ERROR FLAG WORD.
                        ;*                      IF BIT 0 IS SET, A READ ERROR OCCURED.
                        ;*                      IF BIT 1 IS SET, A WRITE ERROR OCCURED.
                        ;*
                        ;* OUTPUTS:     MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER0101" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*********************************************************************

015510                          BGNMSG  ER0101
015510                                                                  ER0101::
015510                          SAVE                    ;SAVE THE GPR CONTENTS.
015510  004567  166262                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

015514  012700  000100          MOV     #BIT06,R0       ;SET-UP THE BIT MAP FOR 'REPORT EXT'D ERROR INFO'
015520  046700  164500          BIC     OPTION,R0       ;TRY AND CLEAR THE FLAG.
015524  001036                  BNE     6$              ;EXIT IF OPTION NOT SELECTED.
                        ;*
                        ; REPORT EXTENDED ERROR INFOMATION
                        ;-

015526  032705  000001          BIT     #BIT0,R5        ;TEST FOR READ ERROR.
015532  001410                  BEQ     2$              ;SKIP READ ERROR MSG IF NO READ ERROR.
015534                          PRINTB  #MSG1           ;PRINT READ ERROR MESSAGE.
015534  012746  015626                                                  MOV     #MSG1,-(SP)
015540  012746  000001                                                  MOV     #1,-(SP)
015544  010600                                                          MOV     SP,R0
015546  104414                                                          TRAP    C$PNTB
015550  062706  000004                                                  ADD     #4,SP
015554  032705  000002          2$:     BIT     #BIT1,R5        ;TEST FOR WRITE ERROR.
015560  001410                  BEQ     4$              ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
                                PRINTB  #MSG2           ;PRINT WRITE ERROR MESSAGE.
015566  012746  015704                                                  MOV     #MSG2,-(SP)
015572  012746  000001                                                  MOV     #1,-(SP)
015574  010600                                                          MOV     SP,R0
015576  104414                                                          TRAP    C$PNTB
015602  062706  000004                                                  ADD     #4,SP
015602                          4$:     PRINTX  #MSG3           ;SUGGEST THAT DHU MAY BE AT WRONG ADDRESS.
015602  012746  015763                                                  MOV     #MSG3,-(SP)
015606  012746  000001                                                  MOV     #1,-(SP)
015612  010600                                                          MOV     SP,R0
015614  104415                                                          TRAP    C$PNTX
015616  062706  000004                                                  ADD     #4,SP
```

```
1882 015622                             6$:     PASS                    ;RESTORE THE GPR CONTENTS.
     015622  004736                                     JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
1883 015624                                     ENDMSG
     015624                                                             L10002:
     015624  104423                                                             TRAP    C$MSG
1884
1885 015626      045     101     102  MSG1::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.%N/
     015631      125     123     040
     015634      124     111     115
     015637      105     055     117
     015642      125     124     040
     015645      124     122     101
     015650      120     040     103
     015653      101     125     123
     015656      105     104     040
     015661      102     131     040
     015664      122     105     101
     015667      104     040     101
     015672      124     124     105
     015675      115     120     124
     015700      056     045     116
     015703      000
1886 015704      045     101     102  MSG2::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.%N/
     015707      125     123     040
     015712      124     111     115
     015715      105     055     117
     015720      125     124     040
     015723      124     122     101
     015726      120     040     103
     015731      101     125     123
     015734      105     104     040
     015737      102     131     040
     015742      127     122     111
     015745      124     105     040
     015750      101     124     124
     015753      105     115     120
     015756      124     056     045
     015761      116     000
1887 015763      045     101     104  MSG3::  .ASCIZ  /%ADHU MAY BE AT THE WRONG UNIBUS ADDRESS.%N%N/
     015766      110     125     040
     015771      115     101     131
     015774      040     102     105
     015777      040     101     124
     016002      040     124     110
     016005      105     040     127
     016010      122     117     116
     016013      107     040     125
     016016      116     111     102
     016021      125     123     040
     016024      101     104     104
     016027      122     105     123
     016032      123     056     045
     016035      116     045     116
     016040      000
1888
1889                                    .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.
                        ;*******************************************************************************

016042                          BGNMSG  ER0201
016042                                                                  ER0201::
016042  004567  165730          SAVE                    JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

016046  012700  000100          MOV     #BIT06,R0       ;TRY TO CLEAR THE
016052  C46700  164146          BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
016056  001025                  BNE     4$              ;EXIT IF FLAG NOT SET.

016060  010102                  MOV     R1,R2
016062  105722          2$:     TSTB    (R2)+           ;CHECK FOR A ZERO BYTE (END OF MESSAGE).
016064  001376                  BNE     2$              ;LOOP UNTIL NEXT MESSAGE IS FOUND.

016066                          PRINTB  #EF0503,R1      ;PRINT THE FIRST MESSAGE.
016066  010146                                                          MOV     R1,-(SP)
016070  012746  004074                                                  MOV     #EF0503,-(SP)
016074  012746  000002                                                  MOV     #2,(SP)
016100  010600                                                          MOV     SP,R0
016102  104414                                                          TRAP    C$PNTB
016104  062706  000006                                                  ADD     #6,SP
016110                          PRINTB  #EF0503,R2      ;PRINT THE SECOND MESSAGE.
016110  010246                                                          MOV     R2,-(SP)
016112  012746  004074                                                  MOV     #EF0503,(SP)
016116  012746  000002                                                  MOV     #2,(SP)
016122  010600                                                          MOV     SP,R0
016124  104414                                                          TRAP    C$PNTB
016126  062706  000006                                                  ADD     #6,SP

016132  004736          4$:     PASS                    JSR     PC,@(SP)+       ;RESTORE THE GPR CONTENTS.
                                                                        ;RETURN TO PREG05 SUBRT.

016134                          ENDMSG
016134                                                                  L10003:
016134  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.
                                ;*********************************************************************************

016136                                  BGNMSG  ER0503
016136                                                                          ER0503::

016136  012700  000100                   MOV    #BIT06,R0       ;TRY TO CLEAR THE
016142  046700  164056                   BIC    OPTION,R0       ;EXT'D ERROR REPORTING FLAG
016146  001011                          BNE     2$              ;EXIT IF FLAG NOT SET.


016150                                  PRINTB  #EF0503,R1      ;PRINT THE MESSAGE.
016150  010146                                                                          MOV     R1,-(SP)
016152  012746  004074                                                                  MOV     #EF0503,-(SP)
016156  012746  000002                                                                  MOV     #2,-(SP)
016162  010600                                                                          MOV     SP,R0
016164  104414                                                                          TRAP    C$PNTB
016166  062706  000006                                                                  ADD     #6,SP

016172                          2$:     ENDMSG
016172                                                                          L10004:
016172  104423                                                                          TRAP    C$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   ER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*************************************************************************

                                BGNMSG  ER0504
016174                                                                          ER0504::
016174
                                 MOV    #BIT06,R0       ;TRY TO CLEAR THE
                                 BIC    OPTION,R0       ;EXTND ERROR REPORTING FLAG
                                BNE     2$              ;EXIT IF FLAG NOT SET.

                                PRINTB  #ER0503,R1      ;PRINT THE FIRST LINE OF THE MESSAGE.
                                                                        MOV     R1,-(SP)
                                                                        MOV     #ER0503,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTB
                                                                        ADD     #6,SP
                                PRINTX  #ER0505,R2      ;PRINT THE NUMBER OF INTS RECEIVED.
                                                                        MOV     R2,-(SP)
                                                                        MOV     #ER0505,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #6,SP

                        2$:     ENDMSG
                                                                L10005:
                                                                        TRAP    C$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.
                        ;*******************************************************************************

016254                          BGNMSG  ER1401
016254                                                                  ER1401::

                        ;
016254  012700  000100          MOV     #BIT06,R0       ;TRY TO CLEAR THE
016260  046700  163740          BIC     OPTION,R0       ;EXTND ERROR REPORTING FLAG
016264  001053                  BNE     60$             ;EXIT IF FLAG NOT SET.
                        ;
016266                          PRINTB  #EF0503,R1      ;REPORT THE ERROR MESSAGE PASSED IN.
016266  010146                                                          MOV     R1,-(SP)
016270  012746  004074                                                  MOV     #EF0503,-(SP)
016274  012746  000002                                                  MOV     #2,-(SP)
016300  010600                                                          MOV     SP,R0
016302  104414                                                          TRAP    C$PNTB
016304  062706  000006                                                  ADD     #6,SP
                        ;
                        ; DETERMINE WHICH ROM VERSION NUMBER(S) ARE MISSING.
                        ;
016310  012705  000143          MOV     #99.,R5         ;GET INVALID ROM NUMBER.
016314  012701  012107          MOV     #EM1405,R1      ;SELECT PROC_1 MESSAGE.
016320  012702  012135          MOV     #EM1407,R2      ;SELECT THE "NOT FOUND" MESSAGE.
016324  120305                  CMPB    R3,R5           ;CHECK PROC_1 ROM VERSION NUMBER.
016326  001402                  BEQ     2$              ;GO REPORT PROC_1 CODE NOT FOUND.
016330  012702  012147          MOV     #EM1408,R2      ;SELECT "FOUND" MESSAGE.
016334  004767  000026  2$:     JSR     PC,50$          ;GO REPORT MESSAGE.
                        ;
016340  012701  012122          MOV     #EM1406,R1      ;SELECT PROC_2 MESSAGE.
016344  012702  012135          MOV     #EM1407,R2      ;SELECT THE "NOT FOUND" MESSAGE.
016350  120405                  CMPB    R4,R5           ;CHECK PROC_2 ROM VERSION NUMBER.
016352  001402                  BEQ     4$              ;GO REPORT PROC_2 CODE NOT FOUND.
016354  012702  012147          MOV     #EM1408,R2      ;SELECT "FOUND" MESSAGE.
016360  004767  000002  4$:     JSR     PC,50$          ;GO REPORT THE MESSAGE.
016364  000413                  BR      60$             ;EXIT.
```

```
2043
2044 016366                        50$:    PRINTX  #EF1402,R1,R2   ;REPORT THE MESSAGE.
     016366  010246                                                                  MOV     R2,-(SP)
     016370  010146                                                                  MOV     R1,-(SP)
     016372  012746  004256                                                          MOV     #EF1402,-(SP)
     016376  012746  000003                                                          MOV     #3,-(SP)
     016402  010600                                                                  MOV     SP,R0
     016404  104415                                                                  TRAP    C$PNTX
     016406  062706  000010                                                          ADD     #10,SP
2045 016412  000207                         RTS     PC              ;RETURN.
2046 016414                        60$:    ENDMSG
     016414                                                                  L10006:
     016414  104423                                                                  TRAP    C$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
                                ;*************************************************************************
016416                                  BGNMSG  ER1601
016416                                                                          ER1601::
016416  012700  000100                    MOV     #BIT06,R0       ;TRY TO CLEAR THE
016422  046700  163576                    BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
016426  001036                          BNE     2$              ;EXIT IF FLAG NOT SET.
016430  016304  002262                    MOV     RMATBB(R3),R4   ;FETCH ADDRESS OF REGISTER NAME MESSAGE.
016434                                  PRINTB  #EF1604,R4,R5   ;REPORT BASIC MESSAGE (REG NAME AND LINE #).
016434  010546                                                                  MOV     R5,-(SP)
016436  010446                                                                  MOV     R4,-(SP)
016440  012746  004451                                                          MOV     #EF1604,-(SP)
016444  012746  000003                                                          MOV     #3,(SP)
016450  010600                                                                  MOV     SP,R0
016452  104414                                                                  TRAP    C$PNTB
016454  062706  000010                                                          ADD     #10,SP
016460                                  PRINTX  #EF1602,R2      ;PRINT THE EXPECTED DATA.
016460  010246                                                                  MOV     R2,(SP)
016462  012746  004345                                                          MOV     #EF1602,-(SP)
016466  012746  000002                                                          MOV     #2,-(SP)
016472  010600                                                                  MOV     SP,R0
016474  104415                                                                  TRAP    C$PNTX
016476  062706  000006                                                          ADD     #6,SP
016502                                  PRINTX  #EF1603,R1      ;PRINT THE ACTUAL DATA.
016502  010146                                                                  MOV     R1,(SP)
016504  012746  004407                                                          MOV     #EF1603,-(SP)
016510  012746  000002                                                          MOV     #2,(SP)
016514  010600                                                                  MOV     SP,R0
016516  104415                                                                  TRAP    C$PNTX
016520  062706  000006                                                          ADD     #6,SP
016524                          2$:     ENDMSG
016524                                                                  1.10007.
```

016524 104423 TRAP C$MSG

PAR

```
                        .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.
                        ;*************************************************************************
016526                          BGNMSG  ER1603
016526                                                                  ER1603::
016526                          SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
016526  004567  165244                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
                        ;
016532  012700  000100          MOV     #BIT06,R0       ;TRY TO CLEAR THE
016536  046700  163462          BIC     OPTION,R0       ;EXTND ERROR REPORTING FLAG
016542  001024                  BNE     2$              ;EXIT IF FLAG NOT SET.
                        ;
                        ;
016544                          PRINTB #EF0503,R1       ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
016544  010146                                                  MOV     R1,-(SP)
016546  012746  004074                                          MOV     #EF0503,-(SP)
016552  012746  000002                                          MOV     #2,-(SP)
016556  010600                                                  MOV     SP,R0
016560  104414                                                  TRAP    C$PNTB
016562  062706  000006                                          ADD     #6,SP
                        ;
016566  016702  165200          MOV     ERRMSG,R2       ;GET THE "TEST MESSAGE".
016572                          PRINTB #EF1601,R2       ;PRINT "TEST ABORTED" MESSAGE.
016572  010246                                                  MOV     R2,-(SP)
016574  012746  004313                                          MOV     #EF1601,-(SP)
016600  012746  000002                                          MOV     #2,(SP)
016604  010600                                                  MOV     SP,R0
016606  104414                                                  TRAP    C$PNTB
016610  062706  000006                                          ADD     #6,SP
                        ;
016614                  2$:     PASS            ;RESTORE THE CONTENTS OF THE GPRS.
016614  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
016616                          ENDMSG
016616                                                          L10010:
016616  104423                                                          TRAP    C$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.
                        ;*************************************************************************

016620                          BGNMSG  ER3001
016620                                                                  ER3001::
                        
016620  012700  000100          MOV     #BIT06,R0       ;TRY TO CLEAR THE
016624  046700  163374          BIC     OPTION,R0       ;EXTD ERROR REPORTING FLAG
016630  001033                  BNE     2$              ;EXIT IF FLAG NOT SET.
                        
016632                          PRINTB  #EF0503,R1      ;PRINT THE FIRST LINE OF THE MESSAGE.
016632  010146                                                          MOV     R1,-(SP)
016634  012746  004074                                                  MOV     #EF0503,-(SP)
016640  012746  000002                                                  MOV     #2,-(SP)
016644  010600                                                          MOV     SP,R0
016646  104414                                                          TRAP    C$PNTB
016650  062706  000006                                                  ADD     #6,SP
016654                          PRINTX  #EF3001,R5      ;REPORT EXPECTED BR LEVEL.
016654  010546                                                          MOV     R5,-(SP)
016656  012746  004546                                                  MOV     #EF3001,-(SP)
016662  012746  000002                                                  MOV     #2,-(SP)
016666  010600                                                          MOV     SP,R0
016670  104415                                                          TRAP    C$PNTX
016672  062706  000006                                                  ADD     #6,SP
016676                          PRINTX  #EF3002,R4      ;REPORT ACTUAL BR LEVEL.
016676  010446                                                          MOV     R4,-(SP)
016700  012746  004615                                                  MOV     #EF3002,-(SP)
016704  012746  000002                                                  MOV     #2,-(SP)
016710  010600                                                          MOV     SP,R0
016712  104415                                                          TRAP    C$PNTX
016714  062706  000006                                                  ADD     #6,SP
                        
016720                  2$:     ENDMSG
016720                                                          L10011:
016720  104423                                                          TRAP    C$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.
                        ;*************************************************************************

                                BGNMSG  ER9004
                                                                                ER9004::
016722  012700  000100            MOV     #BIT06,R0       ;TRY TO CLEAR THE
016726  046700  163272            BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
016732  001040                    BNE     6$              ;EXIT IF FLAG NOT SET.
016734                          PRINTB  #EF0503,#EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
016734  012746  014722                                                        MOV     #EM9014,-(SP)
016740  012746  004074                                                        MOV     #EF0503,-(SP)
016744  012746  000002                                                        MOV     #2,-(SP)
016750  010600                                                                MOV     SP,R0
016752  104414                                                                TRAP    C$PNTB
016754  062706  000006                                                        ADD     #6,SP
016760  005002            CLR     R2              ;CLEAR THE LINE COUNTER.
016762  016703  163474            MOV     ERSMRF,R3       ;GET THE ERROR SUMMARY FLAGS.
016766  005004            CLR     R4              ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
016770  000241    2$:     CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
016772  006003            ROR     R3              ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
016774  103013            BCC     4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
016776                          PRINTX  #EF9010,R2,ERCNTB(R4)
016776  016446  002464                                                        MOV     ERCNTB(R4),-(SP)
017002  010246                                                                MOV     R2,-(SP)
017004  012746  004710                                                        MOV     #EF9010,-(SP)
017010  012746  000003                                                        MOV     #3,-(SP)
017014  010600                                                                MOV     SP,R0
017016  104415                                                                TRAP    C$PNTX
017020  062706  000010                                                        ADD     #10,SP
017024  012405    4$:     MOV     (R4)+,R5        ;INCREMENT THE LINE OFFSET BY 2.
017026  005202            INC     R2              ;INCREMT THE LINE COUNTER.
017030  005703            TST     R3              ;CHECK THE ERROR SUMMARY FLAGS.
017032  001356            BNE     2$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.
017034    6$:     ENDMSG
                                                                        L10012:
017034  104423                                                                TRAP    C$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.
                                ;*******************************************************************************

 017036                                 BGNMSG  ER9007
 017036                                                                         ER9007::

 017036  012700  000100                  MOV    #BIT06,R0       ;TRY TO CLEAR THE
 017042  046700  163156                  BIC    OPTION,R0       ;EXT'D ERROR REPORTING FLAG
 017046  001026                         BNE     2$              ;EXIT IF FLAG NOT SET.

 017050  042703  177760                  BIC    #177760,R3      ;REMOVE ALL BUT LINE # BITS FROM LINE # WORD.
 017054                                 PRINTB  #EF9018,R1,R3   ;REPORT SECONDARY ERROR MESSAGE.
 017054  010346                                                                         MOV     R3,-(SP)
 017056  010146                                                                         MOV     R1,-(SP)
 017060  012746  005150                                                                 MOV     #EF9018,-(SP)
 017064  012746  000003                                                                 MOV     #3,-(SP)
 017070  010600                                                                         MOV     SP,R0
 017072  104414                                                                         TRAP    C$PNTB
 017074  062706  000010                                                                 ADD     #10,SP
 017100                                 PRINTX  #EF9017,R1,R2   ;REPORT THE ACTUAL INCORRECT CODE.
 017100  010246                                                                         MOV     R2,-(SP)
 017102  010146                                                                         MOV     R1,-(SP)
 017104  012746  005074                                                                 MOV     #EF9017,-(SP)
 017110  012746  000003                                                                 MOV     #3,-(SP)
 017114  010600                                                                         MOV     SP,R0
 017116  104415                                                                         TRAP    C$PNTX
 017120  062706  000010                                                                 ADD     #10,SP

 017124                         2$:     ENDMSG
 017124                                                                         L10013:
 017124  104423                                                                         TRAP    C$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.
                        ;*************************************************************************

017126                          BGNMSG  ER9008
017126                                                                  ER9008::
017126  012700  000100                  MOV     #BIT06,R0       ;TRY TO CLEAR THE
017132  046700  163066                  BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
017136  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.
                        ;*
017140  010203                          MOV     R2,R3
017142  000303                          SWAB    R3
017144  042703  177760                  BIC     #177760,R3      ;CALCULATE LINE NUMBER OF CODE.
017150                                  PRINTB  #EF9016,R1,R3   ;REPORT TYPE OF INCORRECT CODE FOUND.
017150  010346                                                                  MOV     R3,-(SP)
017152  010146                                                                  MOV     R1,-(SP)
017154  012746  004777                                                         MOV     #EF9016,-(SP)
017160  012746  000003                                                         MOV     #3,-(SP)
017164  010600                                                                  MOV     SP,R0
017166  104414                                                                  TRAP    C$PNTB
017170  062706  000010                                                         ADD     #10,SP
017174                                  PRINTX  #EF9017,R1,R2   ;REPORT THE ACTUAL INCORRECT CODE.
017174  010246                                                                  MOV     R2,-(SP)
017176  010146                                                                  MOV     R1,-(SP)
017200  012746  005074                                                         MOV     #EF9017,-(SP)
017204  012746  000003                                                         MOV     #3,-(SP)
017210  010600                                                                  MOV     SP,R0
017212  104415                                                                  TRAP    C$PNTX
017214  062706  000010                                                         ADD     #10,SP

017220                          2$:     ENDMSG
017220                                                          L10014:
017220  104423                                                                  TRAP    C$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.
                                        ;**************************************************************************

 017222                                         BGNMSG  ER9101
 017222                                                                                 ER9101::

 017222  012700  000100                          MOV    #BIT06,R0       ;TRY TO CLEAR THE
 017226  046700  162772                          BIC    OPTION,R0       ;EXT'D ERROR REPORTING FLAG
 017232  001012                                 BNE     2$              ;EXIT IF FLAG NOT SET.


 017234                                         PRINTB  #EF9006,R2,R1   ;REPORT THE STRING FOLLOWED BY THE NUMBER.
 017234  010146                                                                                 MOV     R1,-(SP)
 017236  010246                                                                                 MOV     R2,-(SP)
 017240  012746  004664                                                                         MOV     #EF9006,-(SP)
 017244  012746  000003                                                                         MOV     #3,-(SP)
 017250  010600                                                                                 MOV     SP,R0
 017252  104414                                                                                 TRAP    C$PNTB
 017254  062706  000010                                                                         ADD     #10,SP

 017260                                 2$:     ENDMSG
 017260                                                                                 L10015:
 017260  104423                                                                                 TRAP    C$MSG
```

```
2299                                    .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9301 -
2300                                    ;*************************************************************************
2301                                    ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ANY BMP CODES
2302                                    ;*      THAT ARE FOUND IN THE BMP CODE QUEUE, TOGETHER WITH THE THE NUMBER OF
2303                                    ;*      THE TEST THAT WAS EXECUTING AT THE TIME THE BMP CODE WAS LOGGED.
2304                                    ;*      PROVIDED EXTENDED ERROR REPORTING HAS BEEN ENABLED.
2305                                    ;*
2306                                    ;* INPUTS:      R1 - THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
2307                                    ;*               R2 - THE ADDRESS OF THE NEXT EMPTY CELL IN THE QUEUE.
2308                                    ;*
2309                                    ;* OUTPUTS:     THE TEST NUMBER FOLLOWED BY THE BMP CODE ARE PRINTED AT THE
2310                                    ;*               OPERATOR CONSOLE.
2311                                    ;*
2312                                    ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9301" AS THE MESSAGE POINTER
2313                                    ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2314                                    ;*
2315                                    ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2316                                    ;*
2317                                    ;* SUBORDINATE ROUTINES USED: NONE.
2318                                    ;*************************************************************************
2319
2320 017262                                    BGNMSG  ER9301
     017262                                                                                  ER9301::
2321 017262                                    SAVE                    ;SAVE THE GPRS ON THE STACK.
     017262  004567  164510                             JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2322
2323 017266  012700  000100                     MOV     #BIT06,R0       ;TRY TO CLEAR THE
2324 017272  046700  162726                     BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
2325 017276  001064                            BNE     60$             ;EXIT IF FLAG NOT SET.
2326
2327 017300                                    PRINTB  #EF0503,R1      ;REPORT UNEXPECTED BMP CODES FOUND.
     017300  010146                                                                    MOV     R1,-(SP)
     017302  012746  004074                                                            MOV     #EF0503,-(SP)
     017306  012746  000002                                                            MOV     #2,-(SP)
     017312  010600                                                                    MOV     SP,R0
     017314  104414                                                                    TRAP    C$PNTB
     017316  062706  000006                                                            ADD     #6,SP
2328 017322  012703  002500                     MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
2329 017326  012705  015336                     MOV     #EM9302,R5      ;GET THE MESSAGE TO BE REPORTED.
2330 017332  012301                    2$:     MOV     (R3)+,R1        ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
2331 017334  012304                            MOV     (R3)+,R4        ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
2332 017336  004767  000056                    JSR     PC,50$          ;GO REPORT THE BMP CODE.
2333 017342  020302                            CMP     R3,R2           ;CHECK IF ALL CODES HAVE BEEN REPORTED.
2334 017344  103772                            BLO     2$              ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
2335                                    ;*
2336                                    ; CHECK IF OVERFLOW HAS OCCURRED.
2337                                    ;  THE CONDITIONS FOR OVERFLOW ARE: THE POINTER CONTAINS THE ADDRESS OF THE
2338                                    ;  LAST CELL IN THE QUEUE, AND A BMP CODE HAS ALREADY BEEN WRITTEN INTO THAT
2339                                    ;  CELL.
2340                                    ;*
2341 017346  020227  002722                     CMP     R2,#BMPCQE-4    ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
2342 017352  001036                            BNE     60$             ;EXIT IF NOT AT THE LAST LOCATION.
2343 017354  005762  000002                    TST     2(R2)           ;CHECK FOR A BMP CODE IN THE LAST CELL
2344 017360  001433                            BEQ     60$             ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
2345 017362  012301                            MOV     (R3)+,R1        ;GET THE TEST NUMBER OFF THE QUEUE.
2346 017364  011304                            MOV     (R3),R4         ;GET THE BMP CODE OFF THE QUEUE.
2347 017366  012705  015566                    MOV     #EM9303,R5      ;SELECT THE MESSAGE TO BE REPORTED.
```

```
2348 017372                        PRINTX  #EF9302          ;REPORT OVERFLOW CONDITION.
     017372  012746  005333                                                MOV     #EF9302,-(SP)
     017376  012746  000001                                                MOV     #1,-(SP)
     017402  010600                                                        MOV     SP,R0
     017404  104415                                                        TRAP    C$PNTX
     017406  062706  000004                                                ADD     #4,SP
2349 017412  004767  000002        JSR     PC,50$          ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
2350 017416  000414                BR      60$             ;EXIT.
2351
2352 017420                  50$:  PRINTX  #EF9301,R5,R1,R4  ;PRINT THE MESSAGE.
     017420  010446                                                        MOV     R4,-(SP)
     017422  010146                                                        MOV     R1,-(SP)
     017424  010546                                                        MOV     R5,-(SP)
     017426  012746  005255                                                MOV     #EF9301,-(SP)
     017432  012746  000004                                                MOV     #4,-(SP)
     017436  010600                                                        MOV     SP,R0
     017440  104415                                                        TRAP    C$PNTX
     017442  062706  000012                                                ADD     #12,SP
2353 017446  000207                RTS     PC              ;RETURN.
2354 017450                  60$:  PASS                    ;RESTORE THE GPR CONTENTS.
     017450  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2355
2356 017452                        ENDMSG
     017452                                                          L10016:
     017452  104423                                                        TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL SUBROUTINES SECTION

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

```
                        .SBTTL  GLOBAL SUBROUTINE                - 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 HIGHEST SPECIFIED LINE.
                        ;*               THIS ROUTINE DOES NOT READ THE CSR.
                        ;*
                        ;* SUBROUTINES CALLED: NONE.
                        ;-- ************************************************************************

017454                  ALTFLD:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017454  004567  164316                          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.
                        ;-
017460  010400                          MOV     R4,R0           ;CALCULATE THE NEW CONTENTS OF THE
017462  005100                          COM     R0              ;  REGISTER FIELDS WHICH ARE TO BE
017464  040002                          BIC     R0,R2           ;  ALTERRED BY THIS ROUTINE.
017466  016705  162616                  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).
                        ;-
017472  000241                          CLC                     ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
017474  006003          2$:     ROR     R3              ;GET THE LINE SELECT BIT FOR THIS LINE.
017476  103006                  BCC     4$              ;SKIP SETUP IF LINE IS NOT SELECTED.
017500  010577  162536                  MOV     R5,@CSRA        ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
017504  011100                  MOV     (R1),R0         ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
017506  040400                  BIC     R4,R0           ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
017510  050200                  BIS     R2,R0           ;OR IN THE NEW STATES OF THE FIELDS.
017512  010011                  MOV     R0,(R1)         ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
017514  005205          4$:     INC     R5              ;SET LINE NUMBER TO THE NEXT LINE.
017516  005703                  TST     R3              ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
017520  001365                  BNE     2$              ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.
```

```
                ;429
2430 017522                             60$:    PASS                    ;RESTORE GPR'S.
     017522  004736                                     JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
2431 017524  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.
                        ;-- ************************************************************************

017526                  CALMSL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017526  004567  164244                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017532  005067  000210          CLR     62$                     ;CLEAR THE 2ND TIME FLAG.
                        ;*
                        ; SYNCHRONIZE WITH THE LTC.
                        ;-
017536  012705  000001  2$:     MOV     #1,R5                   ;SET OUTER LOOP COUNTER TO 1 LOOP.
                                                        ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE <**
                                                        ;FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <**
017542  005000          CLR     R0                      ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
017544  012767  000001  162572  MOV     #1,TIMER1       ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
017552  005767  162566  4$:     TST     TIMER1          ;CHECK FOR COUNTER HAVING GONE TO ZERO.
017556  001410          BEQ     6$                      ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
017560  005200          INC     R0                      ;COUNT THIS ITERATION OF THE INNER LOOP.
017562  001373          BNE     4$                      ;LOOP IF COUNTER HAS NOT TURNED OVER.
017564  005305          DEC     R5                      ;DECREMENT THE INNER LOOP COUNTER.
017566  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.
                        ;-
017570  005067  162546          CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
017574  000241                  CLC                     ;INDICATE FAILURE FOR RETURN.
017576  000461                  BR      60$             ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
                        ;*
                        ; WE ARE NOW SYNCHRONIZED WITH THE LTC.
                        ; SET UP FOR THE CALIBRATION LOOP.
                        ;
017600  012704  002344  6$:     MOV     #TIMER1,R4      ;WILL TEST TIMER1 IN THE LOOP BELOW.
```

```
2489 017604  005001                        CLR     R1              ;CLEAR THE OUTER LOOP COUNTER.
2490 017606  005002                        CLR     R2              ;INDICATE TO CHECK ALL BITS OF TIMER1.
2491 017610  005003                        CLR     R3              ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2492 017612  012714  000001                MOV     #1,(R4)         ;LOAD TIMER1 WITH COUNT OF 1.
2493
2494 017616  016705  162534          8$:   MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2495 017622  011400                 10$:   MOV     (R4),R0         ;GET THE TIMER1 VALUE.
2496 017624  010067  000120                MOV     R0,64$          ;SAVE WORD (LIKE IN THE REAL LOOP).
2497 017630  040200                        BIC     R2,R0           ;LEAVE ALL THE BITS.
2498 017632  020003                        CMP     R0,R3           ;COMPARE AGAINST ZERO.
2499 017634  000261                        SEC                     ;SET CARRY IN CASE OF SUCCESS.
2500 017636  001406                        BEQ     12$             ;EXIT LOOP IF TIMER1 HAS CLEARED.
2501 017640  005305                        DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2502 017642  001367                        BNE     10$             ;LOOP IF MS NOT UP.
2503 017644  005301                        DEC     R1              ;DECREMENT THE MS TIME COUNT.
2504 017646  001363                        BNE     8$              ;KEEP LOOPING.
2505 017650  004767  000440                JSR     PC,OOPS         ;WE OVERFLOWED, SOMETHING IS WRONG  ABORT.
2506                                ;*
2507                                ; WE HAVE NOW HAVE LOOP COUNT INFORMATION FOR ONE CLOCK TICK.
2508                                ; WE HAVE NEGATIVE OF NUMBER OF OUTER LOOPS IN R1, EACH IS MSLCNT INNER LOOPS.
2509                                ; WE HAVE THE PORTION OF THE LAST OUTER LOOP NOT EXECUTED, IN R5.
2510                                ; NOW WE CALCULATE THE TOTAL NUMBER OF INNER LOOPS EXECUTED.
2511                                ;
2512 017654  005401                12$:   NEG     R1              ;GET NUMBER OF OUTER LOOPS.
2513 017656  016702  162474                MOV     MSLCNT,R2       ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2514 017662  010203                        MOV     R2,R3           ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2515 017664  160502                        SUB     R5,R2           ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2516 017666  010204                        MOV     R2,R4           ;  AND ADD TO ACCUMULATOR LSWORD.
2517 017670  005005                        CLR     R5              ;CLEAR ACCUMULATOR MSWORD.
2518 017672  005301                14$:   DEC     R1              ;CHECK R1 FOR 0 CONDITION
2519 017674  100403                        BMI     16$             ; SKIP MULTIPLICATION IF ZERO
2520 017676  060304                        ADD     R3,R4           ;MULTIPLY NUMBER OF  INNER
2521 017700  005505                        ADC     R5              ; LOOPS PER OUTER LOOP BY
2522 017702  000773                        BR      14$             ;NUMBER OF OUTER LOOPS PERFORMED.
2523                                ;*
2524                                ; DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER LTC TICK.
2525                                ;
2526 017704  016701  162444          16$:   MOV     MSTICK,R1       ;# OF MS PER LTC TICK IS DIVISOR.
2527 017710  010403                        MOV     R4,R3           ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2528 017712  010502                        MOV     R5,R2           ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2529 017714  004767  003140                JSR     PC,UNSDIV       ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK
2530 017720  103402                        BCS     18$             ;BYPASS OOPS IF WE'RE OK.
2531 017722  004767  000366                JSR     PC,OOPS         ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2532 017726  010167  162424          18$:   MOV     R1,MSLCNT       ;SET NEW VALUE FOR MS LOOP COUNT.
2533 017732  005167  000010                COM     62$             ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2534 017736  001277                        BNE     .-$             ;BRANCH IF ONLY ONE ITERATION DONE.
2535 017740  000261                        SEC                     ;SET THE SUCCESS FLAG FOR EXIT.
2536
2537 017742                          60$:   PASS                    ;RESTORE GPRS,
     017742  004736                                JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
2538 017744  000207                        RTS     PC              ; CARRY = SUCCESS FLAG, SET IF SUCCESS.
2539
2540 017746  000000                62$:   .WORD   0               ;2ND CALIBRATION ITERATION FLAGS.
2541 017750  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.
                                ;*************************************************************************

017752                          CKTRAP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
  017752  004567  164020                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017756  005067  162336                  CLR     TP4FLG                  ;CLEAR THE 004 TRAP FLAGS.
017762  011011                          MOV     (R0),(R1)               ;PERFORM THE MOVE IN QUESTION.
017764  005767  162330          ADRPTR:: TST    TP4FLG                  ;CHECK FOR OCCURENCE OF TRAP.
017770  000261                          SEC                             ;INDICATE SUCCESS.
017772  001401                          BEQ     60$                     ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
017774  000241                          CLC                             ;INDICATE FAILURE.
017776                          60$:    PASS                            ;RESTORE GPRS.
  017776  004736                                JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020000  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.
                                ;*************************************************************************

020002                          CLNRST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020002  004567  163770                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;+
                                ; RESET THE DUT.
                                ;  THIS ROUTINE REPORTS ERRORS WITH NUMBERS FROM ERRNBR THRU ERRNBR+2.
                                ;-
020006  004767  001344                  JSR     PC,RESETT               ;RESET THE DUT TO A KNOWN STATE.
020012  103002                          BCC     60$                     ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
                                ;+
                                ; PURGE THE FIFO OF ERROR CODES, SAVE ANY BMP CODES FOUND.
                                ;-
020014  004767  000522                  JSR     PC,PUFIFO               ;PURGE THE FIFO.

020020                          60$:                                    ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
020020                                  PASS                            ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
020020  004736                                          JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
                                                                        ;CARRY BIT:IF CLEAR, THEN ABORT THE TEST.
020022  000207                          RTS     PC
```

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

020024                          CLR16W:: SAVE                           ;SAVE CONTENTS OF GPRS RO THRU R5.
020024  004567  163746                                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
020030  012701  000020                   MOV     #16.,R1                ;SET THE LOOP COUNTER TO 16.
020034  005020                  2$:      CLR     (RO)+                  ;CLEAR A WORD OF MEMORY.
020036  005301                           DEC     R1                     ;COUNT THIS LOOP.
020040  001375                           BNE     2$                     ;LOOP IF NOT 16 WORD CLEARED.
020042                          60$:     PASS                           ;RESTORE GPRS.
020042  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
020044  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.
                        ;-- ************************************************************************

020046                  CNTERR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020046  004567  163724                                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; COUNT THE ERROR ON THE COUNTER FOR THE SPECIFIED LINE.
                        ;-
020052  006305                          ASL     R5              ;FORM WORD OFFSET FROM LINE NUMBER.
020054  016501  002464                  MOV     ERCNTB(R5),R1   ;GET THE PRESENT ERROR COUNT FOR THIS LINE.
020060  005201                          INC     R1              ;COUNT ERROR.
020062  103402                          BCS     2$              ;OVERFLOW?  YES, DON'T UPDATE COUNTER IN TABLE.
020064  010165  002464                  MOV     R1,ERCNTB(R5)   ;UPDATE ERROR COUNTER TABLE ENTRY.
020070  005767  162132          2$:     TST     NDERPT
020074  001411                          BEQ     60$             ;SUMMARYS DISABLED?  YES, EXIT WITH CARRY 0.
020076  020167  162124                  CMP     R1,NDERPT       ;NO.  CHECK FOR ENOUGH ERRORS FOR SUMMARY USE.
020102  101002                          BHI     4$              ;ENOUGH ERRORS TO USE SUMMARY?  YES, GO HANDLE.
020104  000241                          CLC                     ;INDICATE NOT TO USE SUMMARY REPORT YET.
020106  000404                          BR      60$             ;EXIT WITH CARRY 0.
020110  056567  002410  162344  4$:     BIS     BITTBL(R5),ERSMRF       ;SET THE ERROR SUMMARY FLAG FOR LINE.
020116  000261                          SEC                     ;INDICATE TO USE SUMMARY REPORT.
020120                  60$:    PASS                            ;RESTORE GPRS.
020120  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020122  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.
                                ;*****************************************************************************

020124                          DELAY:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
020124  004567  163646                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020130  010401                          MOV     R4,R1           ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
020132  012702  177777                  MOV     #-1,R2          ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
020136  005003                          CLR     R3              ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
020140  012704  020162                  MOV     #62$,R4         ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
020144  004767  000130                  JSR     PC,MSLOOP       ;DELAY THE REQUESTED # OF MS.
020150  103002                          BCC     60$             ;EXIT ROUTINE IF WE TIMED-OUT.)
020152  004767  000136                  JSR     PC,OOPS         ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
020156                          60$:    PASS                    ;RESTORE GPRS.
020156  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020160  000207                          RTS     PC
                                
020162  177777                  62$:    .WORD   -1              ;DUMMY, NON-ZERO WORD.
```

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

020164                  MSLGET:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020164  004567  163606                      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.
                        ;-
020170  005102                  COM     R2              ;GET MASK OF UNUSED BITS.
020172  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.
                        ;-
020174  005701                  TST     R1              ;TEST THE TIME-OUT VALUE FOR ZERO.
020176  001011                  BNE     2$              ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
020200  011400                  MOV     (R4),R0         ;GET THE WORD TO TEST BEFORE EXITING.
020202  010067  000070          MOV     R0,6$           ;SAVE VALUE SO WE CAN RETURN IT.
020206  040200                  BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
020210  020003                  CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
020212  000261                  SEC                     ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
020214  001420                  BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
```

```
2775 020216  000241                        CLC                     ;INDICATE FAILURE (TIME-OUT).
2776 020220  000416                        BR      6$              ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
2777                                ;+
2778                                ; NON-ZERO TIME-OUT VALUE.  LOOP, WAITING FOR CONDITION OR TIME-OUT.
2779                                ;
2780 020222  016705  162130        2$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2781 020226  011400                4$:     MOV     (R4),R0         ;GET THE WORD TO TEST.
2782 020230  010067  000042                MOV     R0,62$          ;SAVE WORD IN CASE THIS IS THE LAST.
2783 020234  040200                        BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
2784 020236  020003                        CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
2785 020240  000261                        SEC                     ;SET CARRY IN CASE OF SUCCESS.
2786 020242  001405                        BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
2787 020244  005305                        DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2788 020246  001367                        BNE     4$              ;LOOP IF MS NOT UP.
2789 020250  005301                        DEC     R1              ;DECREMENT THE MS TIME COUNT.
2790 020252  001363                        BNE     2$              ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
2791 020254  000241                        CLC                     ;CLEAR CARRY, WE TIMED-OUT.
2792                                ;+
2793                                ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME OUT VALUE).
2794                                ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD.  EXIT ROUTINE.
2795                                ;-
2796 020256  016700  000014        6$:     MOV     62$,R0          ;PASS OUT THE LAST READ WORD.
2797 020262                        60$:    PASS    R0,R1           ;RESTORE GPRS, EXCEPT THE FOLLOWING:
     020262  010066  000002                        MOV     R0,ROSLOT(SP)       ;PUT R0 IN STACK SLOT.
     020266  010166  000004                        MOV     R1,R1SLOT(SP)       ;PUT R1 IN STACK SLOT.
     020272  004736                                JSR     PC,@(SP)+           ;RETURN TO PREGOS SUBRT.
2798                                                  ;R0 - LAST READ WORD CHECKED FOR CONDITION.
2799                                                  ;R1   REMAINING TIME (0 IF TIME-OUT OCCURED).
2800 020274  000207                        RTS     PC              ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
2801                                ;+
2802                                ; LOCAL STORAGE.
2803                                ;
2804 020276  000000                62$:    .WORD   0               ;STORAGE FOR THE LAST READ WORD.
```

JFS


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

020300                  MSLOOP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5
020300  004567  163472                          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.
                        ;-
020304  004767  177654          JSR     PC,MSLGET       ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.

020310                  60$:    PASS                            ;RESTORE GPRS,
020310  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
020312  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.
                                  ;-- *******************************************************************************

020314                            OOPS::  SAVE                       ;SAVE CONTENTS OF GPRS R0 THRU R5.
020314  004567  163456                               JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                      ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
020320                                    ERRSF   101,EM0101
020320  104454                                                                          TRAP     C$ERSF
020322  000145                                                                          .WORD    101
020324  020360                                                                          .WORD    EM0101
020326  000000                                                                          .WORD    0
                                      ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
020330                                    PRINTF  #EM0102
020330  012746  020444                                                                  MOV      #EM0102,-(SP)
020334  012746  000001                                                                  MOV      #1,-(SP)
020340  010600                                                                          MOV      SP,R0
020342  104417                                                                          TRAP     C$PNTF
020344  062706  000004                                                                  ADD      #4,SP
020350                            2$:     BREAK                      ;LOOK FOR OPERATOR CONTROL-C INPUT.
020350  104422                                                                          TRAP     C$BRK
020352  000776                            BR      2$                 ;INFINITE LOOP.
020354                            60$:    PASS                       ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
020354  004736                                       JSR     PC,@(SP)+         ;RETURN TO PREG05 SUBRT.
020356  000207                            RTS     PC                 ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.

020360     110     117     123    EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
020363     124     040     103
020366     117     115     120
020371     125     124     105
020374     122     040     110
020377     101     122     104
020402     127     101     122
020405     105     040     117
020410     122     040     123
020413     117     106     124
020416     127     101     122
020421     105     040     102
020424     125     107     040
020427     105     116     103
020432     117     125     116
020435     124     105     122
```

```
        020440   105   104   056
        020443   000
2880    020444   045   116   045   EMO102:: .ASCIZ /%N%APROGRAM HUNG, WAITING FOR A CONTROL-C. <**************%N%N/
        020447   101   120   122
        020452   117   107   122
        020455   101   115   040
        020460   110   125   116
        020463   107   054   040
        020466   127   101   111
        020471   124   111   116
        020474   107   040   106
        020477   117   122   040
        020502   101   040   103
        020505   117   116   124
        020510   122   117   114
        020513   055   103   056
        020516   040   074   052
        020521   052   052   052
        020524   052   052   052
        020527   052   052   052
        020532   052   052   052
        020535   045   116   045
        020540   116   000
2881                                .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.
                        ;**************************************************************************

020542                  PUFIFO::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
020542  004567  163230                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020546  012701  001000          MOV     #512.,R1        ;SET MAXIMUM TRY COUNT OF 512.
020552  016704  161466          MOV     RBUFA,R4        ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
                        2$:     MOV     (R4),R2         ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
020560  100016          BPL     6$              ;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.
                        ;-
020562  012700  070000          MOV     #70000,R0       ;GENERATE A BIT MAP OF CHAR ERROR BITS
020566  040200          BIC     R2,R0           ; WHICH ARE NOT SET FOR CHAR.
020570  001006          BNE     4$              ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
                          ;+
                          ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
                          ;-
020572  012700  000301          MOV     #301,R0         ; CHECK IF BMP.
020576  040200          BIC     R2,R0           ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
020600  001002          BNE     4$              ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
020602  004767  001470          JSR     PC,SAVBMP       ;SAVE BMP CODE ON THE QUEUE.
020606  005301          4$:     DEC     R1              ;DECREMENT THE TRY COUNT.
020610  001362          BNE     2$              ;LOOP TO TRY AGAIN.
020612  000241          CLC                     ;CLEAR CARRY,TO INDICATE FIFO NOT PURGED.
020614  000401          BR      60$             ;EXIT WITH CARRY CLEAR.
020616  000261          6$:     SEC             ;SET CARRY, TO INDICATE FIFO PURGED.
020620                  60$:    PASS                    ;RESTORE GPRS.
020620  004736                          JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
                                                        ;CARRY BIT, SET INDICATES FIFO PURGED.
020622  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.
                                        ;--- *********************************************************************

 020624                                 RDPDR:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
 020624 004567  163146                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
 020630 012767  016416  163136                  MOV     #ER1601,ERRBLK  ;SET UP THE ADDRESS OF THE ERROR REPORT RTN.

                                        ;+
                                        ; DETERMINE WHETHER REGISTER DATA SHOULD BE INVERTED FROM DATA PATTERN.
                                        ;-
 020636 005704                                  TST     R4              ;CHECK THE OPERAND TYPE INDICATOR.
 020640 100001                                  BPL     2$              ;BIC WRITE PERFORMED?  NO, USE STANDARD DATA.
 020642 005102                                  COM     R2              ;YES, INVERT THE DATA PATTERN.
                                        ;+
                                        ; SET UP OUTER LOOP.
                                        ;
 020644 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.
                                        ;-
```

```
2989 020646  010267  000222          4$:     MOV     R2,70$          ;SAVE THE OUTER LOOP DATA PATTERN.
2990 020652  010577  161364                  MOV     R5,@CSRA        ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
2991 020656  010500                          MOV     R5,R0
2992 020660  006300                          ASL     R0
2993 020662  036067  002410  161340          BIT     BITTBL(R0),ACTLNS
2994 020670  001467                        BEQ     16$             ;IS THE LINE ACTIVE?  NO, SKIP THE LINE.
2995 020672  012703  000004                  MOV     @LPR0,R3        ;YES, INITIALIZE REGISTER OFFSET FOR LPR.
2996                                         ;
2997                                         ;       THE INNER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP READS AND COMPARES
2998                                         ;       DATA FROM A DEVICE REGISTER.
2999                                         ;
3000 020676  010204                  6$:     MOV     R2,R4           ;SAVE THE INNER LOOP DATA PATTERN.
3001 020700  046302  002370                  BIC     UNBITB(R3),R2   ;REMOVE UNUSED BITS FROM EXPECTED DATA.
3002 020704  016300  002242                  MOV     DRADRT(R3),R0
3003 020710  005766  000010                  TST     R3SLOT(SP)      ;CHECK THE ACCESS TYPE INDICATOR.
3004 020714  001002                        BNE     8$              ;BYTE ACCESS?  YES, GO PERFORM BYTE READ.
3005 020716  011001                          MOV     (R0),R1         ;NO, PERFORM WORD READ OF DEVICE REGISTER.
3006 020720  000416                        BR      12$
3007 020722  100410                  8$:     BMI     10$             ;LOW BYTE ACCESS?  YES, GO DO LOW BYTE READ.
3008 020724  005200                          INC     R0              ;HIGH BYTE ACCESS.  FORM HIGH BYTE ADDRESS.
3009 020726  111001                          MOVB    (R0),R1         ;READ THE HI BYTE OF THE DUT REGISTER.
3010 020730  000301                          SWAB    R1              ;PUT HI BYTE BACK INTO THE HI BYTE.
3011 020732  042701  000377                  BIC     #377,R1         ;REMOVE THE UNUSED BYTE IN ACTUAL DATA.
3012 020736  042702  000377                  BIC     #377,R2         ;REMOVE THE UNUSED BYTE IN EXPECTED DATA.
3013 020742  000405                        BR      12$
3014 020744  111001                 10$:     MOVB    (R0),R1         ;READ THE LOW BYTE OF THE DUT REGISTER.
3015 020746  042701  177400                  BIC     #177400,R1      ;REMOVE THE UNUSED BYTE.
3016 020752  042702  177400                  BIC     #177400,R2      ;FORM EXPECTED LOW BYTE FOR COMAPARISON.
3017
3018 020756  046301  002370          12$:    BIC     UNBITB(R3),R1   ;REMOVE UNUSED BITS FROM ACTUAL DATA.
3019 020762  020102                          CMP     R1,R2           ;COMPARE ACTUAL AND EXPECTED DATA.
3020 020764  001414                        BEQ     14$             ;ACTUAL = EXPECTED?  YES, SKIP ERROR.
3021 020766  004767  177054                  JSR     PC,CNTERR       ;NO, COUNT THE ERROR, CHECK FOR ERROR SUMMARY.
3022 020772  103411                        BCS     14$             ;USE ERROR SUMMARY?  YES, SKIP ERROR.
3023                                     ;NO, REPORT "BAD BIT(S) IN DEVICE XXXXX REGISTER FOR LINE NN (D)."
3024 020774                              ERROR
     020774  104460                                                                  TRAP    C$ERROR
3025                                     ;
3026                                     ;EXIT THIS ROUTINE AND SET THE "EXIT ON ERROR" FLAG, IF EXTENDED ERROR
3027                                     ;REPORTING HAS NOT BEEN REQUESTED.
3028                                     ;
3029 020776  032767  000100  161220          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3030 021004  001004                        BNE     14$             ;BRANCH IF IT HAS.
3031 021006  012767  000001  161270          MOV     #1,EXOERR       ;SET THE EXIT ON ERROR FLAG.
3032 021014  000425                        BR      60$             ;EXIT THE ROUTINE.
3033
3034 021016  010402                  14$:    MOV     R4,R2           ;RESTORE THE INNER LOOP DATA PATTERN.
3035 021020  004767  000444                  JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
3036 021024  062703  000002                  ADD     #2,R3           ;SET REGISTER OFFSET TO THE NEXT REGISTER.
3037 021030  020327  000006                  CMP     R3,#FIFOSIZ     ;CHECK THAT THIS IS NOT THE FIFOSIZE DATA REG.
3038 021034  001002                        BNE     15$             ;AVOID ALTERING THE OFFSET IF IT ISN'T
3039 021036  062703  000002                  ADD     #2,R3           ;POINT AT THE NEXT REGISTER.
3040 021042  020327  000016          15$:    CMP     R3,#TXBFCO      ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
3041 021046  003713                        BLE     6$              ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
3042                                     ;
3043                                     ; BACK INTO THE OUTER LOOP.  NOW SET UP FOR NEXT LINE.  LOOP IF NOT DONE.
3044                                     ;
```

```
3045 021050  016702  000020          16$:    MOV     70$,R2          ;SET UP TO ROTATE THE DATA PATTERN.
3046 021054  004767  000410                  JSR     PC,ROLDAP       ;ROTATE THE DATA PATTERN.
3047 021060  005205                          INC     R5              ;COUNT THIS LINE
3048 021062  020527  000020                  CMP     R5,#NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
3049 021066  002667                          BLT     4$              ;LOOP IF SOME LINES NOT DONE.
3050
3051 021070                          60$:    PASS                    ;RESTORE GPRS.
     021070  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3052 021074  000207                          RTS     PC
3053
3054 021074  000000                  70$:    .WORD   0               ;STORAGE FOR DATA PATTERN OUTSIDE INNER LOOP.
```

GLOBAL SUBROUTINE REGTST

```
3056                            .SBTTL  GLOBAL SUBROUTINE                - REGTST -
3057                            ;** ************************************************************************
3058                            ;*                              - REGISTERS TEST SUBROUTINE -
3059                            ;*      SUBROUTINE TO TEST THE DEVICE UNDER TEST (DUT) REGISTERS.  THE USED
3060                            ;*      BITS OF THE REGISTERS ARE EITHER ALL CLEARED OR ALL SET AND THEN THE
3061                            ;*      DATA PATTERN IS WRITTEN AND VERIFIED USING EITHER WORD OR BYTE
3062                            ;*      ACCESSES IN READ/WRITE OR READ/MODIFY/WRITE MODE.
3063                            ;*
3064                            ;* INPUTS:      R3 - BYTE INDICATOR (- => LOW, + => HIGH, 0 => BOTH BYTES).
3065                            ;*              R4 - ACCESS MODE (-1 => SET THEN BIC, 1 => CLEAR THEN BIS,
3066                            ;*                      (-2 => SET THEN MOV, +2 CLEAR THEN MOV).
3067                            ;*              ERRNBR - SET UP WITH INITIAL ERROR NUMBER.
3068                            ;*
3069                            ;* OUTPUTS:     GPR$0 - GPR SAVE AREA 0 IS DESTROYED.
3070                            ;*              DEVICE UNDER TEST REGISTERS ARE WRITTEN.
3071                            ;*              ERROR MESSAGES MAY BE PRINTED AT THE OPERATORS CONSOLE.
3072                            ;*
3073                            ;* CALLING SEQUENCE:    JSR     PC,REGTST
3074                            ;*
3075                            ;* COMMENTS:    THIS ROUTINE LOOP 16 TIMES WRITING THE SAME DATA PATTERN
3076                            ;*              ROTATED LEFT ONCE EACH ITERATION.
3077                            ;*              THIS ROUTINE CAN REPORT ERRORS INITIAL ERRNBR THRU INITIAL+2.
3078                            ;*
3079                            ;* SUBORDINATE ROUTINES CALLED: RDPDR,ROLDAP,SWAPO,WDPDR
3080                            ;-- ************************************************************************
3081
3082 021076                     REGTST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     021076  004567  162674                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3083                            ;*
3084                            ; SET UP THE GPRS FOR THE WRITTING OF THE DATA PATTERN.
3085                            ;-
3086 021102  012705  000020                      MOV     #16.,R5         ;SET UP LOOP COUNTER TO COUNT 16 ITERATIONS.
3087 021106  012702  167410                      MOV     #167410,R2      ;INITIALIZE THE DATA PATTERN.
3088 021112  032704  000001                      BIT     #1,R4           ;TEST FOR R/W ACCESS.
3089 021116  001001                             BNE     2$              ;R/M/W ACCESS?  YES, R4 IS ALL SET UP.
3090 021120  005004                              CLR     R4              ;NO, INDICATE R/W ACCESS.
3091 021122                     2$:
3092                            ;*
3093                            ; SET UP THE GPRS FOR THE CLEARING OR SETTING OF ALL THE USED BITS.
3094                            ;-
3095 021122  010400                              MOV     R4,R0           ;PASS OPERATION TYPE INDICATOR AROUND SWAPO.
3096 021124  004767  001272                      JSR     PC,SWAPO        ;GET ALTERNATE GPR SET IN R1 THRU R5.
3097 021130  016701  162634                      MOV     ERRNBR,R1       ;SAVE THE INITIAL ERROR NUMBER.
3098 021134  010004                              MOV     R0,R4
3099 021136  005404                              NEG     R4              ;SET UP OP TYPE FOR CLEARING OR SETTING.
3100 021140  005002                              CLR     R2              ;SET UP CLEAR WRITE PATTERN.
3101 021142  026627  000012  000002              CMP     R4,SLOT(SP),#2  ;TEST FOR CLEAR THEN MOV TEST SEQUENCE.
3102 021150  001401                             BEQ     4$              ;CLEAR THEN MOV?  YES, LEAVE WRITE PAT CLEAR.
3103 021152  005102                              COM     R2              ;NO, SET ALL BITS OF WRITE PATTERN.
3104 021154  005003                     4$:      CLR     R3              ;INDICATE THAT WORD ACCESSES SHOULD BE USED.
3105 021156  005000                              CLR     R0              ;SET ALTERNATE BYTE EXPECTED DATA PAT TO CLEAR.
3106 021160  026627  000012  177776              CMP     R4,SLOT(SP),#-2 ;TEST FOR SET THEN MOV TEST SEQUENCE.
3107 021166  001001                             BNE     6$              ;SET THEN MOV?  YES, LEAVE ALT BYTE PAT CLEAR.
3108 021170  005100                              COM     R0              ;NO, SET ALT BYTE EXPECTED DATA PAT TO ALL 1'S.
3109 021172  004767  001224              6$:     JSR     PC,SWAPO        ;RESTORE SWAPPED GPR VALUES TO R1 THRU R5.
3110                            ;*
3111                            ; START OF DATA PATTERN LOOP.
```

```
3112                                    ;-
3113 021176                             8$:
3114                                        ;+
3115                                        ; SET OR CLEAR ALL THE USED BITS OF THE DEVICE REGISTERS FOR ALL LINES.
3116                                        ; VERIFY THAT ALL THE BITS WERE SET OR CLEARED CORRECTLY.
3117                                        ;-
3118 021176  004767  001220                    JSR     PC,SWAPO        ;GET ALTERNATE GPRS FOR SETTING INTIAL STATES.
3119 021202  004767  002136                    JSR     PC,WDPDR        ;GO CLEAR ALL USED REGISTER BITS, ALL LINES.
3120 021206  010167  162556                     MOV    R1,ERRNBR       ;SET UP ERROR NUMBER TO INITIAL ERRNBR.
3121 021212  004767  177406                    JSR     PC,RDPDR        ;VERIFY ALL USED REGISTER BITS, ALL LINES.
3122                                    ;+
3123                                    ;EXIT THIS ROUTINE IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING
3124                                    ;HAS BEEN REQUESTED,  I.E.  EXOERR IS NON-ZERO.
3125                                    ;-
3126 021216  005767  161062                     TST    EXOERR          ;HAS AN ERROR BEEN FOUND ?
3127 021222  001035                            BNE     60$             ;EXIT THIS ROUTINE IF IT HAS.
3128
3129 021224  004767  001172                    JSR     PC,SWAPO        ;RESTORE MAIN GPRS CONTENTS.
3130                                        ;+
3131                                        ; WRITE DATA PATTERNS, ALL LOWER BYTE USED BITS, ALL REGISTERS, ALL LINES.
3132                                        ; VERIFY THAT THE DATA PATTERN WAS WRITTEN CORRECTLY.
3133                                        ;-
3134 021230  004767  002110                    JSR     PC,WDPDR        ;WRITE DATA PATTERN TO DEVICE REGISTERS.
3135 021234  005267  162530                     INC    ERRNBR          ;SET ERROR NUMBER TO INITIAL+1.
3136 021240  004767  177360                    JSR     PC,RDPDR        ;VERIFY DATA PATTERN IN ALTERRED BYTE(S).
3137                                    ;+
3138                                    ;EXIT THIS ROUTINE IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING
3139                                    ;HAS BEEN REQUESTED.
3140                                    ;-
3141 021244  005767  161034                     TST    EXOERR          ;HAS AN ERROR BEEN FOUND ?
3142 021250  001022                            BNE     60$             ;EXIT THIS ROUTINE IF IT HAS.
3143
3144 021252  005703                             TST    R3              ;CHECK THE BYTE INDICATOR.
3145 021254  001414                            BEQ     10$             ;WORD ACCESS?  YES, SKIP SECOND BYTE CHECK.
3146                                        ;+
3147                                        ; CHECK THAT THE ALTERNATE (UNMODIFIED) BYTE IS CLEAR OR SET AS EXPECTED.
3148                                        ;-
3149 021256  010201                             MOV    R2,R1           ;SAVE THE DATA PATTERN.
3150 021260  010002                             MOV    R0,R2           ;GET THE ALTERNATE BYTE EXPECTED DATA.
3151 021262  005403                             NEG    R3              ;INDICATE THAT OTHER BYTE IS TO BE CHECKED.
3152 021264  005267  162500                     INC    ERRNBR          ;SET ERROR NUMBER TO INITIAL+2.
3153 021270  004767  177330                    JSR     PC,RDPDR        ;VERIFY DATA PATS IN OTHER BYTES OF REGISTERS.
3154                                    ;+
3155                                    ;EXIT THIS ROUTINE IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING
3156                                    ;HAS BEEN REQUESTED.
3157                                    ;-
3158 021274  005767  161004                     TST    EXOERR          ;HAS AN ERROR BEEN FOUND ?
3159 021300  001006                            BNE     60$             ;EXIT THIS ROUTINE IF IT HAS.
3160
3161 021302  005403                             NEG    R3              ;RESTORE BYTE INDICATOR.
3162 021304  010102                             MOV    R1,R2           ;RESTORE DATA PATTERN.
3163                                        ;+
3164                                        ; PEPARE THE NEXT DATA PATTERN AND LOOP IF NOT DONE.
3165                                        ;-
3166 021306  004767  000156             10$:    JSR     PC,ROLDAP       ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
3167 021312  005305                             DEC    R5              ;COUNT THIS ITERATION OF THE LOOP.
3168 021314  003330                            BGT     8$              ;ALL PATTERNS DONE?  NO, LOOP.
```

```
3169                                                            ;YES, RESTORE ERROR NUMBER AND EXIT.
3170 021316  016767  161126  162444  60$:    MOV     GPRSOB,ERRNBR   ;GET THE ERROR NUMBR FROM GPR SWAP STORAGE.
3171 021324                                  PASS                    ;RESTORE GPRS.
     021324  004736                                  JSR     PC,@(SP)+               ;RETURN TO REGOS SUBRT.
3172 021326  000207                          RTS     PC
```

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

021330                  REPSMR:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
021330  004567  162442                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021334  005767  161122          TST     ERSMRF          ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
021340  001404                  BEQ     60$             ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
                        ;*
                        ; WE HAVE SOME ERROR SUMMARIES TO REPORT.
                        ;-
021342  012767  016722  162424          MOV     #ER9004,ERRBLK  ;SELECT ERROR REPORTING ROUTINE.
                        ;*
                        ; REPORT
                        ;     "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
                        ;-
                            ERROR
021350                                                                  TRAP    C$ERROR
021350  104460
                        60$:    PASS                     ;RESTORE GPRS.
021352                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
021352  004736
021354  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.
                        ;**************************************************************************

021356                  RESETT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
021356  004567  162414          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021362  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.
                        ;-
021366  016704  160650          MOV     CSRA,R4         ;GET THE ADDRESS OF THE DUT'S CSR.
021372  030214                  BIT     R2,(R4)         ;CHECK STATE OF MASTER RESET BIT.
021374  001406                  BEQ     2$              ;DON'T DELAY IF MR IS ALREADY CLEAR.
021376  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
021400  012701  011610          MOV     #5000.,R1       ;PASS TIME-OUT VALUE OF 5 SECONDS.
021404  004767  176554          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
021410  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.
                        ;
021412  010277  160624  2$:     MOV     R2,@CSRA        ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
021416  004767  000722          JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
                          ;*
                          ;   SET SELF TEST TIME OUT OF 5 SECONDS, AND WAIT FOR M.R TO CLEAR.
                          ;   IF TIME OUT OCCURS, THEN REPORT THE FATAL ERROR AND PASS-OUT THE ABORT
                          ;   TEST INDICATOR.
                          ;-
021422  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
021424  012701  011610          MOV     #5000.,R1       ;PASS TIME-OUT VALUE OF 5 SECONDS.
021430  004767  176530          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
021434  103410                  BCS     6$              ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
```

```
                                        ;*
                                        ; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET,TEST ABORTED".
                                        ; INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
                                        ;
021436  012701  012221          4$:     MOV     #EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
021442  012767  016526  162324          MOV     #ER1603,ERRBLK  ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
                                        ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
                                        ; "TEST ABORTED"
021450                                  ERROR                   ;               >>>>> ERROR <<<<<
021450  104460                                                                    TRAP    C$ERROR
021452  000241                          CLC                     ;INDICATE TEST IS TO BE ABORTED.
021454  000403                          BR      60$             ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
                                        ;*
                                        ; CLEAR TX AND RX INTERRUPT ENABLE STATUS FLAGS IN IESTAT.
                                        ; EXIT WITH CONTINUE TEST INDICATOR SET (IE,CARRY SET).
                                        ;
021456  005067  160626          6$:      CLR    IESTAT          ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
021462  000261                           SEC                    ;INDICATE SUCCESS, CONTINUE TEST.

021464                          60$:    PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
021464  004736                                    JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
                                                                ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
021466  000207                          RTS     PC
```

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

021470                  ROLDAP::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
021470  004567  162302                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021474  005702                  TST     R2              ;CHECK MSB, AND CLEAR CARRY.
021476  100001                  BPL     2$              ;BRANCH IF CLEAR.
021500  000261                  SEC                     ;SET CARRY IF MSB SET
021502  006102          2$:     ROL     R2              ;ROTATE DATA PATTERN LEFT
021504                  60$:    PASS    R2              ;RESTORE GPRS,EXCEPT
021504  010266  000006                  MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
021510  004736                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                        ;R2 - CONTAINS THE ROTATED DATA PATTERN
021512  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.
                                        ;-- ************************************************************************

 021514                                 RSTRPT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
 021514  004567  162256                                 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.
                                        ;-
 021520  005003                                 CLR     R3              ;CLEAR THE CODE COUNTER.
 021522  016705  162242                         MOV     ERRNBR,R5       ;SAVE ERRNBR FOR RESTORATION LATER.
 021526  017702  160512                 2$:     MOV     @RBUFA,R2       ;READ A CHAR FROM THE DUT FIFO.
 021532  100422                                 BMI     4$              ;SKIP ERROR IF DATA.VALID SET FOR CHAR.

                                        ;+
                                        ; WE EXPECT A SELFTEST CODE, BUT THIS FIFO SLOT IS EMPTY.
                                        ;-
 021534  010567  162230                         MOV     R5,ERRNBR       ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
 021540  012701  015127                         MOV     #EM9018,R1      ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
 021544  012767  017036  162222                 MOV     #ER9007,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
                                        ;+
                                        ; REPORT ERROR WITH NUMBER INITIAL ERRNBR.
                                        ;    "NO SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET.
                                        ;
 021552                                         ERROR                   ;                       >>>>> ERROR <<<<<.
 021552  104460                                                                                         TRAP    C$ERROR
                                        ;+
                                        ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
                                        ;-
 021554  032767  000100  160442                 BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
 021562  001003                                 BNE     3$              ;AVOID SET THE FLAG IF IT HAS.
 021564  012767  000001  160512                 MOV     #1,EXOERR       ;SET THE EXIT ON ERROR FLAG
```

```
3378                                    ;+
3379                                    ; INIDICATE "SUCCESS" (BECAUSE FIFO IS PURGED), AND EXIT THIS ROUTINE.
3380                                    ;-
3381 021572  000261                3$:      SEC                     ;SET SUCCESS FLAG.
3382 021574  000167  000406              JMP     60$                 ;EXIT ROUTINE.
3383                                    ;+
3384                                    ; DETERMINE IF THIS IS NOT A SELFTEST CODE.
3385                                    ;-
3386 021600  012700  070001        4$:      MOV     #70001,R0          ;GENERATE BIT MAP OF ANY CLEAR ERROR BITS OR
3387 021604  040200                       BIC     R2,R0              ;  BIT 0 WHICH ARE CLEAR.
3388 021606  001042                      BNE     8$                  ;GO TO REPORT ERROR IF THIS IS NOT A TEST CODE.
3389                                    ;+
3390                                    ; WE HAVE A TEST CODE (EITHER BMP OR SELFTEST CODE).
3391                                    ; DETERMINE WHAT TYPE OF CODE WE HAVE.
3392                                    ;-
3393 021610  032702  000200                BIT     #BIT7,R2           ;TEST ROM VERSION CODE INDICATOR BIT.
3394 021614  001462                      BEQ     10$                 ;SKIP ERRORS IF SELFTEST ROM VERSION CODE.
3395 021616  120227  000203                CMPB    R2,#203            ;CHECK IF SKIP SELF TEST CODE.
3396 021622  001457                      BEQ     10$                 ;SKIP ERROR REPORT IF SKIP SELF TEST CODE FOUND
3397 021624  120227  000201                CMPB    R2,#201            ;CHECK IF NULL CODE PRESENT.
3398 021630  001454                      BEQ     10$                 ;SKIP ERROR REPORT IF SELF TEST NULL CODE.
3399 021632  012700  000300                MOV     #300,R0            ;TEST CODE TYPE BITS FOR BOTH CODE
3400 021636  040200                       BIC     R2,R0              ;  TYPE BITS SET (INDICATING BMP CODE).
3401 021640  001003                      BNE     6$                  ;IF IT IS NOT A BMP CODE GO REPORT ERROR.
3402 021642  004767  000430              JSR     PC,SAVBMP           ;SAVE THE BMP CODE ON THE QUEUE.
3403 021646  000445                      BR      10$                 ;GO GET THE NEXT CHARACTER FROM THE FIFO.
3404                                    ;+
3405                                    ; WE HAVE A SELFTEST ERROR CODE.
3406                                    ;-
3407 021650  010567  162114        6$:      MOV     R5,ERRNBR          ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
3408 021654  005267  162110                INC     ERRNBR             ;CALCULATE INITIAL ERROR NUMBER PLUS 1.
3409 021660  012701  015154                MOV     #EM9020,R1         ;PASS ERROR MESSAGE INFO TO ER9008 ROUTINE.
3410 021664  012767  017126  162102        MOV     #ER9008,ERRBLK     ;SELECT PROPER ERROR REPORT ROUTINE.
3411                                    ;+
3412                                    ; REPORT ERROR WITH NUMBER INITIAL ERRNBR + 1.
3413                                    ;    "UNEXPECTED SELFTEST ERROR CODE FOR LINE NN IN FIFO AFTER RESET:"
3414                                    ;-
3415 021672                               ERROR                   ;               >>>>> ERROR <<<<<.
     021672  104460                                                                   TRAP    C$ERROR
3416                                    ;+
3417                                    ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3418                                    ;-
3419 021674  032767  000100  160322        BIT     #BIT06,OPTION      ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3420 021702  001027                      BNE     10$                 ;AVOID SET THE FLAG IF IT HAS AND GO TO
3421                                                                 ;THE END OF THE LOOP.
3422 021704  012767  000001  160372        MOV     #1,EXOERR          ;SET THE "EXIT ON ERROR" FLAG
3423 021712  000534                      BR      50$                 ;EXIT THE ROUTINE WITH FAILURE SINCE THE FIFO
3424                                                                 ;IS NOT PRUGED.
3425                                    ;+
3426                                    ; WE HAVE A NON-SELFTEST CODE (EITHER BMP CODE OR DATA CHAR).
3427                                    ;
3428 021714  010567  162050        8$:      MOV     R5,ERRNBR          ;RESTORE ERROR NUMBER TO INTITIAL VALUE.
3429 021720  062767  000002  162042        ADD     #2,ERRNBR          ;CALCULATE INITIAL ERROR NUMBER PLUS 2.
3430 021726  012701  015137                MOV     #EM9019,R1         ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
3431 021732  012767  017036  162034        MOV     #ER9007,ERRBLK     ;SELECT PROPER ERROR REPORT ROUTINE.
3432                                    ;+
3433                                    ; REPORT ERROR WITH NUMBER INITIAL ERRNBR + 2.
```

```
                                        ;   "NON-SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET."
                                        ;-
3436 021740                                 ERROR                   ;                   >>>>> ERROR <<<<<.
     021740  104460                                                                                   TRAP    C$ERROR
3437                                    ;+
3438                                    ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3439                                    ;-
3440 021742  032767  000100  160254        BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3441 021750  001004                        BNE     10$             ;AVOID SET THE FLAG IF IT HAS AND GO TO
3442                                                                ;THE END OF THE LOOP.
3443 021752  012767  000001  160324        MOV     #1,EXOERR       ;SET THE "EXIT ON ERROR" FLAG
3444 021760  000511                        BR      50$             ;EXIT THE ROUTINE WITH FAILURE.
3445
3446                                    ;+
3447                                    ; END OF LOOP, LOOP IF NOT ALL CHARS HAVE BEEN READ FROM THE FIFO.
3448                                    ;-
3449 021762  005203                10$:     INC     R3              ;SET CODE COUNTER FOR NEXT ITERATION OF LOOP.
3450 021764  020327  000010                CMP     R3,#8.          ;TEST FOR ALL CODES READ.
3451 021770  002656                        BLT     2$              ;LOOP IF NOT CHARS READ FROM FIFO.
3452                                    ;+
3453                                    ; PURGE THE FIFO UNTIL DATA.VALID IS CLEAR OR UNTIL TOO MANY CHARS ARE READ.
3454                                    ;-
3455 021772  012704  000022                MOV     #18.,R4         ;INITIALIZE THE CHARACTER COUNTER.
3456 021776  010567  161766                MOV     R5,ERRNBR       ;GET INITIAL VALUE OF THE ERROR NUMBER.
3457 022002  062767  000003  161760        ADD     #3,ERRNBR       ;CALCULATE ERROR NUMBER OF NEXT ERROR.
3458 022010  012767  017126  161756        MOV     #ER9008,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3459 022016  017702  160222            12$:MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE DUT FIFO.
3460 022022  000261                        SEC                     ;INDICATE SUCCESS IN CASE DATA.VALID IS CLEAR.
3461 022024  100070                        BPL     60$             ;EXIT ROUTINE WITH SUCCESS IF DATA.VALID CLEAR.
3462                                    ;+
3463                                    ; WE HAVE A CHARACTER.
3464                                    ; DETERMINE IF CHAPACTER IS A DATA CHARACTER.
3465                                    ;-
3466 022026  012700  070000                MOV     #70000,R0       ;TEST BITS 12 THRU 14 OF THE
3467 022032  040200                        BIC     R2,R0           ;  CODE READ FROM THE DUT FIFO.
3468 022034  001403                        BEQ     14$             ;SKIP THIS ERROR IF CODE IS NOT A DATA CHAR.
3469                                    ;+
3470                                    ; WE HAVE AN UNEXPECTED DATA CHARACTER: SET UP AND GO TO REPORT ERROR.
3471                                    ;-
3472 022036  012701  015200                MOV     #EM9022,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3473 022042  000423                        BR      22$             ;GO TO REPORT THIS ERROR.
3474                                    ;+
3475                                    ; WE HAVE AN UNEXPECTED CODE.
3476                                    ; DETERMINE IF THE CODE IS A MODEM STATUS CODE.
3477                                    ;-
3478 022044  032702  000001            14$:BIT     #BIT0,R2        ;TEST MODEM STATUS INDICATOR BIT OF CODE.
3479 022050  001003                        BNE     16$             ;SKIP THIS ERROR IF NOT MODEM STATUS CODE.
3480                                    ;+
3481                                    ; WE HAVE A MODEM STATUS CODE; SET UP AND GO TO REPORT ERROR.
3482                                    ;-
3483 022052  012701  015217                MOV     #EM9023,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3484 022056  000415                        BR      22$             ;GO TO REPORT THIS ERROR.
3485                                    ;+
3486                                    ; WE HAVE AN ONBOARD TEST CODE.
3487                                    ; DETERMINE IF THIS CODE IS A BMP CODE.
3488                                    ;-
3489 022060  032702  000200            16$:BIT     #BIT7,R2        ;TEST THE ROM VERSION BIT OF THE CODE.
```

```
3490 022064  001404                        BEQ     18$             ;GOTO SET UP FOR SELFTEST CODE IF ROM VERSION.
3491 022066  012700  000300                 MOV     #300,R0
3492 022072  040200                         BIC     R2,R0           ;TEST THE ERROR TYPE BITS OF THE CODE.
3493 022074  001403                        BEQ     20$             ;SKIP THIS ERROR IF BMP CODE.
3494                                    ;+
3495                                    ; WE HAVE A SELFTEST CODE: SET UP AND GO TO REPORT ERROR.
3496                                    ;-
3497 022076  012701  015241          18$:    MOV     #EM9024,R1      ;SELECT ERROR MSG INFO FOR ER0806 ROUTINE.
3498 022102  000403                        BR      22$             ;GO TO REPORT THIS ERROR.
3499                                    ;+
3500                                    ; WE HAVE A BMP CODE: SAVE IT ON THE QUEUE.
3501                                    ;-
3502 022104  004767  000166          20$:    JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
3503 022110  000411                        BR      24$             ;
3504                                    ;+
3505                                    ; REPORT THE ERROR WITH ERROR NUMBER OF INITIAL ERRNBR + 3.
3506                                    ;    "UNEXPECTED XXX XXXX FOR LINE NN IN FIFO AFTER RESET:"
3507                                    ;-
3508 022112                          22$:    ERROR                   ;               >>>>> ERROR <<<<<.
     022112  104460                                                                      TRAP    C$ERROR
3509                                    ;+
3510                                    ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3511                                    ;-
3512 022114  032767  000100  160102         BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3513 022122  001004                        BNE     24$             ;AVOID SETTING THE FLAG IF IT HAS AND GO TO
3514                                                                ;THE END OF THE LOOP.
3515 022124  012767  000001  160152         MOV     #1,EXOERR       ;SET THE "EXIT ON ERROR" FLAG
3516 022132  000424                        BR      50$             ;EXIT THE ROUTINE WITH FAILURE.
3517
3518                                    ;+
3519                                    ; END OF LOOP.
3520                                    ; COUNT THE CHARACTER WE JUST RECEIVED, AND CHECK FOR TOO MANY RECEIVED.
3521                                    ;-
3522 022134  005304                  24$:    DEC     R4              ;COUNT THIS CHARACTER.
3523 022136  001327                        BNE     12$             ;LOOP IF NOT TOO MANY CHARACTERS PURGED.
3524                                    ;+
3525                                    ; WE READ TOO MANY VALID CHARACTERS WHILE TRYING TO PURGE THE FIFO.
3526                                    ; REPORT ERROR AND EXIT WITHOUT SUCCESS.
3527                                    ;    "FIFO WILL NOT PURGE (DATA.VALID STUCK SET), REMAINDER OF TEST SKIPPED."
3528                                    ;-
3529 022140  012701  015016                 MOV     #EM9017,R1      ;SELECT PROPER ERROR MESSAGE.
3530 022144  010567  161620                 MOV     R5,ERRNBR       ;GET INITIAL ERROR NUMBER.
3531 022150  062767  000004  161612         ADD     #4,ERRNBR       ;CALCULATE INITIAL ERRNBR + 4.
3532 022156  012767  016136  161610         MOV     #ER0503,ERRBLK  ;SELECT PROPER ERROR REPORT ROUTINE.
3533                                        ;PRINT ERROR REPORT.
3534 022164                                 ERROR                   ;               >>>>> ERROR <<<<<.
     022164  104460                                                                      TRAP    C$ERROR
3535                                    ;+
3536                                    ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3537                                    ;-
3538 022166  032767  000100  160030         BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3539 022174  001003                        BNE     50$             ;AVOID SETTING THE FLAG IF IT HAS.
3540 022176  012767  000001  160100         MOV     #1,EXOERR       ;SET THE "EXIT ON ERROR" FLAG
3541
3542 022204  000241                  50$:    CLC                     ;CLEAR THE SUCCESS FLAG.
3543
3544 022206                          60$:    PASS                    ;RESTORE GPRS.
```

```
     022206  004736                    JSR     PC,@(SP)+              ;RETURN TO PREG05 SUBRT.
3545 022210  000207            RTS     PC                             ; CARRY - SUCCESS FLAG (SET IF FIFO IS PURGED).
```

C8


```
3547                                     .SBTTL  GLOBAL SUBROUTINE                  - RXIEO -
3548                                     ;** *******************************************************************
3549                                     ;*                        RECEIVER INTERRUPT DISABLE
3550                                     ;*      THIS ROUTINE IS USED TO DISABLE RECEIVER INTERRUPTS IN THE DHU11.
3551                                     ;*
3552                                     ;* INPUTS:        NONE.
3553                                     ;*
3554                                     ;* OUTPUTS:       THE RX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
3555                                     ;*                IESTST  CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3556                                     ;*                ENABLE BITS.
3557                                     ;*
3558                                     ;* CALLING SEQUENCE:     JSR     PC,RXIEO
3559                                     ;*
3560                                     ;* COMMENTS:      THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3561                                     ;*                THE DUT CSR ARE DESTROYED.
3562                                     ;*
3563                                     ;* SUBORDINATE ROUTINES CALLED: NONE.
3564                                     ;** *******************************************************************
3565 022212  010046                      RXIEO::  MOV     R0,-(SP)          ;SAVE CONTENTS OF R0 ON THE STACK.
3566 022214                                       GETPRI  -(SP)             ;SAVE PROCESSOR PRIORITY ON STACK.
     022214  104440                                                                                    TRAP    C$GPRI
     022216  010046                                                                                    MOV     R0,-(SP)
3567 022220                                       SETPRI  #PRI07            ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
     022220  012700  000340                                                                            MOV     #PRI07,R0
     022224  104441                                                                                    TRAP    C$SPRI
3568 022226  042767  137777  160054               BIC     #137777,IESTAT    ;CLEAR RX.INT.ENBL BIT IN IESTAT.
3569 022234  016777  160050  160000               MOV     IESTAT,@CSRA      ;DISABLE RX INTERRUPTS.
3570 022242                                       SETPRI  (SP)+             ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
     022242  012600                                                                                    MOV     (SP)+,R0
     022244  104441                                                                                    TRAP    C$SPRI
3571 022246  012600                               MOV     (SP)+,R0          ;RESTORE R0.
3572 022250  000207                               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.
                                        ;-- ****************************************************************************

 022252  052767  000100  160030  RXIE1::  BIS     #BIT06,IESTAT      ;SET RX.INT.ENBL BIT IN IESTAT.
 022260  042767  137677  160022           BIC     #137677,IESTAT     ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
 022266  016777  160016  157746           MOV     IESTAT,@CSRA       ;ENABLE RX INTERRUPTS.
 022274  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.
                        ;-- ************************************************************************

022276                  SAVBMP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
022276  004567  161474                 JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
022302  016704  160216          MOV     BMPCQP,R4               ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
022306  116724  160012          MOVB    TSTNUM,(R4)+            ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
022312  005204                  INC     R4                      ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
022314  042702  177400          BIC     #177400,R2              ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
022320  010224                  MOV     R2,(R4)+                ;SAVE THE BMP CODE ON THE QUEUE.
022322  020427  002726          CMP     R4,#BMPCQE              ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
022326  103402                  BLO     2$                      ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
022330  162704  000004          SUB     #4,R4                   ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
022334  010467  160164  2$:     MOV     R4,BMPCQP               ;SAVE THE POINTER.

022340                  60$:    PASS                            ;RESTORE GPRS.
022340  004736                         JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
022342  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.
                        ;-- ************************************************************************

022344                  SKPSTS:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
022344  004567  161426                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022350  012704  000012          MOV     #10.,R4         ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
022354  004767  175544         JSR     PC,DELAY        ;DELAY FOR 10 MILLI-SECONDS.
                        ;+
                        ; WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
                        ;-
022360  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.
022364  012703  052525          MOV     #52525,R3       ;INITIALISE THE SKIP SELF-TEST CODE.
022370  005301          4$:     DEC     R1              ;SELECT THE NEXT SET OF DEVICE REGISTERS.
022372  016704  157644          MOV     CSRA,R4         ;GET THE ADDRESS OF THE CSR OF THE DUT.
022376  010124                  MOV     R1,(R4)+        ;SELECT A BANK OF DUT REGISTERS.
022400  010324          6$:     MOV     R3,(R4)+        ;WRITE THE CODE TO A DUT REGISTER.
022402  020467  157652          CMP     R4,TXBFCA       ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
022406  103774                  BLO     6$              ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
022410  032701  000017          BIT     #17,R1          ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
022414  001365                  BNE     4$              ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.

022416                  60$:    PASS                    ;RESTORE GPRS.
022416  004736                          JSR     R5,@(SP)+               ;RETURN TO PREG05 SUBRT.
022420  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.
                                ;-- ************************************************************************

022422  010046                  SWAPO::  MOV     R0,-(SP)        ;SAVE THE CONTENTS OF R0.
                                ;*
                                ; LOAD THE STACK FROM THE GPRS.
                                ;-
022424  010146                          MOV     R1,-(SP)        ;SAVE THE CONTENTS OF R1.
022426  010246                          MOV     R2,-(SP)        ;SAVE THE CONTENTS OF R2.
022430  010346                          MOV     R3,-(SP)        ;SAVE THE CONTENTS OF R3.
022432  010446                          MOV     R4,-(SP)        ;SAVE THE CONTENTS OF R4.
022434  010546                          MOV     R5,-(SP)        ;SAVE THE CONTENTS OF R5.
                                ;*
                                ; LOAD THE GPRS FROM THE GPR SAVE AREA 0.
                                ;-
022436  012700  002450                  MOV     #GPRS0B,R0      ;GET THE BASE ADDRESS OF GPR SAVE AREA 0.
022442  012001                          MOV     (R0)+,R1        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 1.
022444  012002                          MOV     (R0)+,R2        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 2.
022446  012003                          MOV     (R0)+,R3        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 3.
022450  012004                          MOV     (R0)+,R4        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 4.
022452  012005                          MOV     (R0)+,R5        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 5.
                                ;*
                                ; LOAD THE GPR SAVE AREA 0 FROM THE STACK.
                                ;-
022454  012640                          MOV     (SP)+, (R0)     ;LOAD GPR SAVE AREA 0 WORD 5 WITH SAVED R5.
022456  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 4 WITH SAVED R4.
022460  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 3 WITH SAVED R3.
022462  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 2 WITH SAVED R2.
022464  012640                          MOV     (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 1 WITH SAVED R1.

022466  012600                          MOV     (SP)+,R0        ;RESTORE THE INITIAL VALUE OF R0.

022470  000207                         RTS     PC
```

```
                                        .SBTTL  GLOBAL SUBROUTINE                - TSABRT -
                                        ;++ ************************************************************************
                                        ;*                      - TEST ABORT ROUTINE -
                                        ;*      THIS SUBROUTINE IS USED WHEN A NON-TEST RELATED ERROR HAS BEEN FOUND
                                        ;*      DURING THE EXECUTION OF THE CURRENT TEST.
                                        ;*      IT IS USED TO INFORM THE OPERATOR THAT THE CURRENT TEST HAS BEEN
                                        ;*      ABORTED.
                                        ;*
                                        ;* INPUTS:      ERRMSG - CONTAINS THE NAME OF THE CURRENT TEST.
                                        ;*              ERRNBR - CONTAINS THE CURRECT ERROR NUMBER.
                                        ;*              THE REMAINDER OF THE ERRTBL IS CORRECTLY INITIALISED.
                                        ;*
                                        ;* OUTPUTS:     MESSAGES ARE REPORTED TO THE OPERATOR.
                                        ;*
                                        ;* CALLING SEQUENCE:    JSP     PC,TSABRT
                                        ;*
                                        ;* COMMENTS:
                                        ;*
                                        ;* SUBORDINATE ROUTINES CALLED: ER1603.
                                        ;-- ************************************************************************

022472                                  TSABRT:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
022472  004567  161300                                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022476  012701  022514                          MOV     #2$,R1          ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
022502  012767  016526  161264                  MOV     #ER1603,ERRBLK  ;SET-UP THE ERROR REPORTING ROUTINE.
022510                                          ERROR                   ;               >>>>> ERROR <<<<<.
022510  104460                                                                                    TRAP    C$ERROR
022512  000432                                  BR      60$             ;
022514     040     116     117  2$:             .ASCIZ / NON-RELATED TEST ERROR FOUND DURING TEST EXECUTION/
022517     116     055     122
022522     105     114     101
022525     124     105     104
022530     040     124     105
022533     123     124     040
022536     105     122     122
022541     117     122     040
022544     106     117     125
022547     116     104     040
022552     104     125     122
022555     111     116     107
022560     040     124     105
022563     123     124     040
022566     105     130     105
022571     103     125     124
022574     111     117     116
022577     000
                                        .EVEN
022600                                  60$:    PASS                     ;RESTORE GPRS.
022600  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
022602  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.
                                    ;** ************************************************************************

022604                              TXDSBL:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     022604  004567  161166                             JSR      R5,PREG05         ;CALL REGISTER SAVE SUBRT.
022610  010500                               MOV     R5,R0             ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
022612  012701  000001                       MOV     #BIT0,R1          ;INITIALIZE THE SELECTED LINE BIT MASK.
022616  016702  157434                       MOV     TXAD2A,R2         ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
022622  005202                               INC     R2                ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
022624  012703  000020                       MOV     #NUMLNS,R3        ;GET MAXIMUM LINE NUMBER PLUS ONE.
022630  016704  157454                       MOV     IESTAT,R4         ;GET THE STATES OF THE INT ENABLE BITS.
022634  005005                               CLR     R5                ;LOG POSSIBLE TX DISABLED ON ALL LINES.
                                    ;*
                                    ; SELECT EVERY LINE IN TURN, AND LOG THE STATE OF EACH TX.ENABLE BIT.
                                    ;
022636  010477  157400              2$:      MOV     R4,@CSRA          ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
022642  105712                               TSTB    (R2)              ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
022644  100001                               BPL     4$                ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
022646  050105                               BIS     R1,R5             ;LOG TX ENABLE BIT SET FOR SELECTED LINE.

                                     ;*
                                     ;  CLEAR TX.ENBLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX DISABLE
                                     ;  LINE BIT MAP.
                                     ;
022650  030100                      4$:      BIT     R1,R0             ;CHECK STATE OF DISABLE LINE BIT MAP.
022652  001402                               BEQ     6$                ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
022654  142712  000200                       BICB    #BIT7,(R2)        ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
022660  005204                      6$:      INC     R4                ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
022662  006301                               ASL     R1                ;SHIFT BIT MAP FOR NEXT LINE.
022664  005303                               DEC     R3                ;DECREMENT LINE NUMBER.
022666  001363                               BNE     2$                ;LOOP TO CHECK NEXT LINE.

022670                              60$:     PASS    R5                ;RESTORE GPRS,EXCEPT
     022670  010566  000014                             MOV      R5,R5SLOT(SP)             ;PUT R5 IN STACK SLOT.
     022674  004736                                     JSR      PC,@(SP)+                 ;RETURN TO PREG05 SUBRT.
                                                                       ;R5   PREVIOUS STATES OF ALL TX.ENABLE BITS.
022676  000207                               RTS     PC
```

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

3838 022700             TXENBL:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
     022700  004567  161072              JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3839 022704  010500             MOV     R5,R0           ;COPY BIT MAP OF LINES TO ENABLE.
3840 022706  012701  000001     MOV     #BIT0,R1        ;INITIALIZE THE SELECTED LINE BIT MASK.
3841 022712  016702  157340     MOV     TXAD2A,R2       ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
3842 022716  005202             INC     R2              ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
3843 022720  012703  000020     MOV     #NUMLNS,R3      ;GET MAXIMUM LINE NUMBER.
3844 022724  016704  157360     MOV     IESTAT,R4       ;GET THE STATES OF THE INT ENABLE BITS.
3845 022730  005005             CLR     R5              ;CLEAR TX.ENABLE BIT LOG OF DISABLED LINES.
3846                            ;*
3847                            ; SELECT EVERY LINE IN TURN,AND LOG ANY TX.ENBLE BIT THAT IS CLEAR.
3848                            ;-
3849 022732  010477  157304     2$:     MOV     R4,@CSRA        ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
3850 022736  105712             TSTB    (R2)            ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
3851 022740  100401             BMI     4$              ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
3852 022742  050105             BIS     R1,R5           ;LOG TX ENABLE BIT CLEAR FOR SELECTED LINE.
3853                            ;*
3854                            ;  SET TX.ENBLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX ENABLE
3855                            ;  LINE BIT MAP.
3856                            ;-
3857 022744  030100             4$:     BIT     R1,R0           ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
3858 022746  001402             BEQ     6$              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
3859 022750  152712  000200     BISB    @BIT7,(R2)      ;ENABLE TRANSMISSION ON SELECTED LINE.
3860 022754  005204             6$:     INC     R4              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
3861 022756  006301             ASL     R1              ;SHIFT BIT MAP FOR NEXT LINE.
3862 022760  005303             DEC     R3              ;DECREMENT LINE NUMBER.
3863 022762  001363             BNE     2$              ;LOOP TO CHECK NEXT LINE.
3864
3865 022764             60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
     022764  010566  000014                     MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
     022770  004736                             JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3866                                                         ;R5   LINE BIT MAP CORRESPONDING TO THE
3867                                                         ; PREVIOUS LINES THAT WERE DISABLED.
3868 022772  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.
                                        ;-- ************************************************************************
022774  010046                          TXIEO:: MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
022776                                          GETPRI  -(SP)           ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
022776  104440                                                                  TRAP    C$GPRI
023000  010046                                                                  MOV     R0,-(SP)
023002                                          SETPRI  #PRIO7          ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
023002  012700  000340                                                          MOV     #PRIO7,R0
023006  104441                                                                  TRAP    C$SPRI
023010  042767  177677  157272                 BIC     #177677,IESTAT  ;CLEAR TX.INT.ENBL BIT IN IESTAT.
023016  016777  157266  157216                 MOV     IESTAT,@CSRA    ;DISABLE TX INTERRUPTS.
023024                                         SETPRI (SP)+            ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
023024  012600                                                                  MOV     (SP)+,R0
023026  104441                                                                  TRAP    C$SPRI
023030  012600                                 MOV     (SP)+,R0        ;RESTORE R0.
023032  000207                                 RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - TXIE1 -
                                ;++ ************************************************************************
                                ;*                          - TRANSMITTER INTERRUPT ENABLE -
                                ;*      THIS ROUTINE IS USED TO ENABLE TRANSMITTER INTERRUPTS IN THE DHU11.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      THE TX INT.ENBL BIT IS SET IN THE DUT CSR.
                                ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
                                ;*               ENABLE BITS.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TXIE1
                                ;*
                                ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*               THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

023034  052767  040000  157246  TXIE1::  BIS     @BIT14,IESTAT   ;SET TX.INT.ENBL BIT IN IESTAT.
023042  042767  137677  157240          BIC     @137677,IESTAT  ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
023050  016777  157234  157164          MOV     IESTAT,@CSRA    ;ENABLE TX INTERRUPTS.
023056  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.
                        ;-- ************************************************************************

023060                  UNSDIV:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023060  004567  160712                 JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
                        ;-
023064  010204                          MOV     R2,R4           ;GET MSW OF DIVIDEND FOR SUBTRACT.
023066  160104                          SUB     R1,R4           ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
023070  103403                          BCS     2$              ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
023072  012701  177777                  MOV     #-1,R1          ;SET QUOTIENT TO ALL ONES (177777).
023076  000442                          BR      60$             ;EXIT WITH CARRY CLEAR.
                        ;+
                        ; SET UP COUNTERS AND VARIOUS WORKING GPRS.
                        ;-
023100  005004                  2$:     CLR     R4              ;CLEAR THE LSW OF THE DIVISOR.
023102  000241                          CLC                     ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
023104  006001                          ROR     R1              ; DIVISOR BY
023106  006004                          ROR     R4              ; 2(UNSIGNED)
023110  012700  000020                  MOV     #16.,R0         ;SET UP INITIAL SHIFT COUNT TO 16.
                        ;+
                        ; THE SUBTRACT AND SHIFT LOOP.
                        ;-
023114  010246                  4$:     MOV     R2,-(SP)        ;SAVE MSWORD OF DIVIDEND.
023116  010346                          MOV     R3,-(SP)        ;SAVE LSWORD OF DIVIDEND.
023120  160403                          SUB     R4,R3           ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
023122  005602                          SBC     R2              ;MSWORD DIVIDEND - BORROW.
023124  103402                          BCS     6$              ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
023126  160102                          SUB     R1,R2           ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
023130  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.
                                ;-
023132  012603                  6$:     MOV     (SP)+,R3        ;RESTORE LSWORD OF DIVIDEND.
023134  012602                          MOV     (SP)+,R2        ;RESTORE MSWORD OF DIVIDEND.
```

```
3977 023136  000401                        BR      10$             ;GOTO SHIFT 1 INTO THE QUOTIENT.
3978                                    ;+
3979                                    ; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
3980                                    ; COMPLEMENTED LATER).   CARRY IS CLEAR.
3981                                    ;-
3982 023140  012626                 8$:    MOV     (SP)+,(SP)+     ;POP THE SAVED DIVIDEND OFF OF THE STACK.
3983                                    ;+
3984                                    ; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.
3985                                    ;-
3986 023142  006105                 10$:   ROL     R5              ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
3987 023144  000241                        CLC                     ;DIVIDE THE
3988 023146  006001                        ROR     R1              ;  DEVISOR BY
3989 023150  006004                        ROR     R4              ;  2 (UNSIGNED).
3990 023152  005300                        DEC     R0              ;COUNT THIS SHIFT AND SUBTRACT.
3991 023154  001357                        BNE     4$              ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
3992 023156  005105                        COM     R5              ;GET QUOTIENT FROM INVERTED QUOTIENT.
3993                                    ;+
3994                                    ; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.
3995                                    ;-
3996 023160  000241                        CLC                     ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
3997 023162  006103                        ROL     R3              ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
3998 023164  103402                        BCS     12$             ;IF CARRY FROM SHIFT, ROUND UP.
3999 023166  160403                        SUB     R4,R3           ;SUBTRACT DIVISOR FROM DIVIDEND.
4000 023170  103403                        BCS     14$             ;IF BORROW, DON'T ROUND UP.
4001                                    ;+
4002                                    ; ROUND UP, EXTRA SUBTRACT WENT.
4003                                    ;-
4004 023172  005205                 12$:   INC     R5              ;INCREMENT THE QUOTIENT BY ONE.
4005 023174  001001                        BNE     14$             ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
4006 023176  005305                        DEC     R5              ;DON'T LET ROUNDING CAUSE OVERFLOW.
4007                                    ;+
4008                                    ; ALL DONE, PASS QUOTIENT AND EXIT.
4009                                    ;-
4010 023200  010501                 14$:   MOV     R5,R1           ;PASS QUOTIENT BACK IN R1.
4011 023202  000261                        SEC                     ;INDICATE NO OVERFLOW.
4012
4013 023204                         60$:   PASS    R1                      ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
     023204  010166  000004                                 MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
     023210  004736                                         JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
4014                                                                 ;R1   16 BIT, UNSIGNED QUOTIENT.
4015 023212  000207                        RTS     PC              ;CARRY - SET INDICATES NO OVERFLOW (SUCCESS).
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - WAIBIC -
                        ;** ************************************************************************
                        ;*                      - WAIT FOR BIT CLEAR ROUTINE -
                        ;*      THIS SUBROUTINE WAITS FOR THE SPECIFIED BIT TO BECOME CLEAR.  IF THE
                        ;*      SPECIFIED BIT GOES TO A CLEAR STATE WITHIN THE SPECIFIED TIME-OUT
                        ;*      PERIOD A SUCCESS INDICATION IS RETURNED BY THIS ROUTINE.
                        ;*      THE LAST VALUE WHICH IS READ LOOKING FOR THE CONDITION IS RETURNED TO
                        ;*      ALLOW THE USE OF THIS ROUTINE TO LOOK FOR DESTRUCTIVE READ CONDITIONS.
                        ;*
                        ;* INPUTS:      R1 - TIME-OUT VALUE AND BIT NUMBER INDICATION:
                        ;*                BITS 15 THRU 12 - NUMBER OF BIT TO TEST (RANGE 0 THRU 15).
                        ;*                BITS 11 THRU 0  - TIME-OUT VALUE IN MILLI-SECONDS (4095 MAX).
                        ;*              R2 - ADDRESS OF WORD CONTAINING THE BIT TO TEST.
                        ;*                   MSLCNT.
                        ;*
                        ;* OUTPUTS:     R2   THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
                        ;*              CARRY - SUCCESS FLAG (CARRY SET IF BIT CLR BEFORE TIME-OUT).
                        ;*
                        ;* CALLING SEQUENCE:    MOV     #130040,R1      ;PASS BIT 11 (13 OCTAL) AND
                        ;*                                              ;  32 (40 OCTAL) MS DELAY.
                        ;*                      MOV     #LABEL,R2       ;TEST BIT IN WORD AT "LABEL".
                        ;*                      JSR     PC,WAIBIC       ;WAIT 32 MS FOR BIT 11 TO CLR.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: MSLGET.
                        ;** ************************************************************************
023214                  WAIBIC:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
023214  004567  160556                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023220  010204                  MOV     R2,R4                   ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
023222  010102                  MOV     R1,R2
023224  042701  170000          BIC     #170000,R1              ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
023230  042702  007777          BIC     #7777,R2                ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
023234  000302                  SWAB    R2                      ;PUT LINE NUMBER FIELD IN LSBYTE.
023236  006202                  ASR     R2                      ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
023240  006202                  ASR     R2                      ;  POSITION TO USE IT AS A WORD TABLE OFFSET
023242  006202                  ASR     R2                      ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
023244  016202  002410          MOV     BITTBL(R2),R2           ;GET BIT MAP OF LINE TO TEST FROM TABLE.
023250  005003                  CLR     R3                      ;INDICATE THAT THE BIT SHOULD BE CLR.
023252  004767  174706          JSR     PC,MSLGET               ;WAIT FOR THE BIT TO BE CLR WITHIN TIME OUT.
                                                                ;  CARRY IS CORRECT UPON MSLGET RETURN.
023256  010002                  MOV     R0,R2                   ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
023260                  60$:    PASS    R2                      ;RESTORE GPRS, EXCEPT THE FOLLOWING:
023260  010266  000006                          MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
023264  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;  R2   LAST VALUE READ LOOKING FOR CONDITION.
023266  000207                  RTS     PC                      ;  CARRY - SUCCESS FLAG (SET IF BIT FOUND CLR).
```

```
                                .SBTTL  GLOBAL SUBROUTINE                   - WAIBIS -
                                ;-- *************************************************************
                                ;*                      - WAIT FOR BIT SET ROUTINE -
                                ;*      THIS SUBROUTINE WAITS FOR THE SPECIFIED BIT TO BECOME SET.  IF THE
                                ;*      SPECIFIED BIT GOES TO A SET STATE WITHIN THE SPECIFIED TIME-OUT
                                ;*      PERIOD A SUCCESS INDICATION IS RETURNED BY THIS ROUTINE.
                                ;*      THE LAST VALUE WHICH IS READ LOOKING FOR THE CONDITION IS RETURNED TO
                                ;*      ALLOW THE USE OF THIS ROUTINE TO LOOK FOR DESTRUCTIVE READ CONDITIONS.
                                ;*
                                ;* INPUTS:        R1 - TIME-OUT VALUE AND BIT NUMBER INDICATION:
                                ;*                  BITS 15 THRU 12 - NUMBER OF BIT TO TEST (RANGE 0 THRU 15).
                                ;*                  BITS 11 THRU 0  - TIME-OUT VALUE IN MILLI-SECONDS (4095 MAX).
                                ;*                R2 - ADDRESS OF WORD CONTAINING THE BIT TO TEST.
                                ;*                        MSLCNT.
                                ;*
                                ;* OUTPUTS:       R2 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
                                ;*                CARRY - SUCCESS FLAG (CARRY SET IF BIT SET BEFORE TIME-OUT).
                                ;*
                                ;* CALLING SEQUENCE:      MOV     #130040,R1      ;PASS BIT 11 (13 OCTAL) AND
                                ;*                                                ;  32 (40 OCTAL) MS DELAY.
                                ;*                        MOV     #LABEL,R2       ;TEST BIT IN WORD AT "LABEL".
                                ;*                        JSR     PC,WAIBIS       ;WAIT 32 MS FOR BIT 11 TO SET.
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: MSLGET.
                                ;-- *************************************************************

023270                          WAIBIS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023270  004567  160502                                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023274  010204                           MOV     R2,R4                  ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
023276  010102                           MOV     R1,R2
023300  042701  170000                   BIC     #170000,R1             ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
023304  042702  007777                   BIC     #7777,R2               ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
023310  000302                           SWAB    R2                     ;PUT LINE NUMBER FIELD IN LSBYTE.
023312  006202                           ASR     R2                     ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
023314  006202                           ASR     R2                     ;  POSITION TO USE IT AS A WORD TABLE OFFSET
023316  006202                           ASR     R2                     ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
023320  016202  002410                   MOV     BITTBL(R2),R2          ;GET BIT MAP OF LINE TO TEST FROM TABLE.
023324  010203                           MOV     R2,R3                  ;INDICATE THAT THE BIT SHOULD BE SET.
023326  004767  174632                   JSR     PC,MSLGET              ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
                                                                        ;  CARRY IS CORRECT UPON MSLGET RETURN.
023332  010002                           MOV     R0,R2                  ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
023334                          60$:     PASS    R2                     ;RESTORE GPRS, EXCEPT THE FOLLOWING:
023334  010266  000006                                  MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
023340  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                        ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
023342  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.
                                      ;;  ******************************************************************

 023344                               WDPDR:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
         023344  004567  160426                       JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                      ;*
                                      ; SET UP OUTER LOOP WHICH WRITES THE DATA PATTERN TO EACH LINE'S REGISTERS
                                      ;
 023350  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.
                                      ;
 023352  010204                       2$:     MOV     R2,R4           ;SAVE THE OUTER LOOP DATA PATTERN.
```

```
4165 023354  010577  156662                    MOV     R5,@CSRA          ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
4166 023360  006305                            ASL     R5                ;TURN LINE NUMBER INTO A WORD OFFSET.
4167 023362  036567  002410  156640            BIT     BITTBL(R5),ACTLNS
4168 023370  001456                           BEQ     20$               ;LINE ACTIVE?  NO, SKIP THIS LINE.
4169 023372  012701  000004                    MOV     #LPRO,R1          ;YES, INITIALIZE THE REGISTER OFFSET.
4170                                  ;+
4171                                  ;       THE INNER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP WRITES DATA TO A
4172                                  ;       DEVICE REGISTER.
4173                                  ;-
4174 023376  010200                  4$:      MOV     R2,R0
4175 023400  046100  002370                    BIC     UNBTTB(R1),R0     ;CLEAR BIT FIELDS FOR UNUSED REGISTER BITS.
4176 023404  016103  002242                    MOV     DRADRT(R1),R3     ;GET THE ADDRESS OF THE DEVICE REGISTER.
4177 023410  005766  000010                    TST     R3SLOT(SP)        ;CHECK THE OPERAND TYPE INDICATOR.
4178 023414  003402                           BLE     6$                ;HIGH BYTE?  NO, SKIP HIGH BYTE ADDRESS SET UP.
4179 023416  005203                            INC     R3                ;YES, SET THE REG ADDRESS TO THE HIGH BYTE.
4180 023420  000300                            SWAB    R0                ;MOVE HIGH BYTE DATA INTO THE LOW BYTE.
4181 023422  005766  000010          6$:       TST     R3SLOT(SP)        ;CHECK THE OPERAND TYPE INDICATOR.
4182 023426  001412                           BEQ     12$               ;WORD ACCESS?  YES, GO PERFORM WORD ACCESS.
4183                                      ;+
4184                                      ;PERFORM BYTE ACCESS TO THE SPECIFIED BYTE OF THE SPECIFIED REGISTER.
4185                                      ;-
4186 023430  005766  000012                    TST     R4SLOT(SP)        ;NO, CHECK THE ACCESS TYPE INDICATOR.
4187 023434  100403                           BMI     8$                ;USE BIC?  YES, GO PERFORM BICB INSTRUCTION.
4188 023436  001404                           BEQ     10$               ;USE MOV?  YES, GO PERFORM MOVB INSTRUCTION.
4189 023440  150013                            BISB    R0,(R3)           ;NEITHER.  PERFORM BISB ACCESS TO REGISTER.
4190 023442  000415                           BR      18$
4191 023444  140013                  8$:       BICB    R0,(R3)           ;PERFORM BICB ACCESS TO REGISTER.
4192 023446  000413                           BR      18$
4193 023450  110013                 10$:       MOVB    R0,(R3)           ;PERFORM MOVB ACCESS TO REGISTER.
4194 023452  000411                           BR      18$
4195                                      ;+
4196                                      ;PERFORM WORD ACCESS TO THE SPECIFIED REGISTER.
4197                                      ;-
4198 023454  005766  000012          12$:      TST     R4SLOT(SP)        ;CHECK THE ACCESS TYPE INDICATOR.
4199 023460  100403                           BMI     14$               ;USE BIC?  YES, GO PERFORM BIC INSTRUCTION.
4200 023462  001404                           BEQ     16$               ;USE MOV?  YES, GO PERFORM MOV INSTRUCTION.
4201 023464  050013                            BIS     R0,(R3)           ;NEITHER.  PERFORM BIS ACCESS TO REGISTER.
4202 023466  000403                           BR      18$
4203 023470  040013                 14$:       BIC     R0,(R3)           ;PERFORM BIC ACCESS TO REGISTER.
4204 023472  000401                           BR      18$
4205 023474  010013                 16$:       MOV     R0,(R3)           ;PERFORM MOV ACCESS TO REGISTER.
4206                                      ;+
4207                                      ; PREPARE THE DATA PATTERN AND OFFSET FOR THE NEXT REGISTER ON THIS LINE.
4208                                      ;-
4209 023476  004767  175766          18$:      JSR     PC,ROLDAP         ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
4210 023502  062701  000002                    ADD     #2,R1             ;INCREMENT OFFSET FOR NEXT REGISTER.
4211 023506  020127  000006                    CMP     R1,#FSLSO         ;CHECK IF THIS IS THE FIFOSIZE/DATA REG
4212 023512  001002                           BNE     19$               ;AVOID ALTERING THE OFFSET IF IT ISN'T.
4213 023514  062701  000002                    ADD     #2,R1             ;AVOID TESTING THESE REGISTERS.
4214 023520  020127  000016          19$:      CMP     R1,#TXBFCO        ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
4215 023524  003724                           BLE     4$                ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
4216                                      ;+
4217                                      ; BACK INTO THE OUTER LOOP.  NOW SET UP FOR NEXT LINE.  LOOP IF NOT DONE.
4218                                      ;-
4219 023526  010402                 20$:      MOV     R4,R2             ;SET UP TO ROTATE THE DATA PATTERN.
4220 023530  004767  175734                    JSR     PC,ROLDAP         ;ROTATE THE DATA PATTERN.
4221 023534  006205                            ASR     R5                ;CONVERT BACK TO LINE NUMBER FROM WORD OFFSET.
```

```
4222 023536  005205                    INC     R5              ;COUNT THIS LINE.
4223 023540  020527  000020            CMP     R5,#NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
4224 023544  002702                    BLT     2$              ;LOOP IF SOME LINES NOT DONE.
4225
4226 023546                    60$:    PASS                    ;RESTORE GPRS.
     023546  004736                            JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
4227 023550  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.
                        ;-- ************************************************************************

023552                  WTWLNC:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023552  004567  160220                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
                        ;-
023556  016701  156470          MOV     LNCTRA,R1       ;SET UP THE REGISTER ADDRESS PARAMETER.
023562  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
023564  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
023566  012704  177777          MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                        ;*
                        ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
                        ;-
023572  004767  173656          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

023576                  60$:    PASS                            ;RESTORE GPRS.
023576  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023600  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.
                        ;-- ************************************************************************

023602                  WTWLPR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023602  004567  160170                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
                        ;-
023606  016701  156434          MOV     LPRA,R1         ;SET UP THE REGISTER ADDRESS PARAMETER.
023612  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
023614  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
023616  012704  177777          MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                        ;*
                        ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
                        ;-
023622  004767  173626          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

023626                  60$:    PASS                    ;RESTORE GPRS.
023626  004736                                  JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
023630  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
                                        ;** ************************************************************************

023632                                  CACHRX::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
023632  004567  160140                                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023636  016701  156452                          MOV     RXINTC,R1       ;GET THE RECEIVER INTERRUPT COUNT
023642  005201                               INC     R1              ;INCREMENT THE COUNT
023644  102001                               BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED
023646  005301                               DEC     R1              ;RESET THE COUNT TO 177777
023650  010167  156440                  2$:     MOV     R1,RXINTC       ;SAVE NEW COUNT VALUE
023654                                  60$:    PASS                    ;RESTORE GPRS.
023654  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023656  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
                                ;-- ************************************************************************

023660                          CACHTX::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
023660  004567  160112                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023664  016701  156436                MOV     TXINTC,R1       ;GET THE TRANSMISSION INTERRUPT COUNT
023670  005201                        INC     R1              ;INCREMENT THE COUNT
023672  102001                        BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED
023674  005301                        DEC     R1              ;RESET THE COUNT TO 177777
023676  010167  156424        2$:     MOV     R1,TXINTC       ;SAVE NEW COUNT VALUE
023702                          60$:    PASS                    ;RESTORE GPRS.
023702  004736                                JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023704  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.
                        ;-- ****************************************************************************

023706  005767  156432  CLKINT:: TST     TIMER1          ;CHECK FOR TIMER1 AT ZERO.
023712  001402                   BEQ     2$              ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
023714  005367  156424           DEC     TIMER1          ;DECREMENT TIME COUNT.
023720  005767  156422  2$:      TST     TIMER2          ;CHECK FOR TIMER2 AT ZERO.
023724  001402                   BEQ     4$              ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
023726  005367  156414           DEC     TIMER2          ;DECREMENT TIME COUNT.
023732  005367  156412  4$:      DEC     TIMER3          ;DECREMENT THE BREAK COUNT.
023736  001006                   BNE     60$             ;EXIT IF NOT TIME TO CALL BREAK.
023740  016767  156406  156402   MOV     BCOUNT,TIMER3   ;SET UP TIME TILL NEXT BREAK.
023746  010046                   MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 FROM BREAK MACRO.
023750                           BREAK                   ;CHECK FOR OPERATOR CONTROL/C.
023750  104422                                                           TRAP    C$BRK
023752  012600                   MOV     (SP)+,R0        ;RESTORE CONTENTS OF R0.
023754  000002           60$:    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.
                        ;-- ************************************************************************

4430 023756             RXBRRT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
     023756  004567  160014              JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
4431 023762  017700  156256              MOV     @RBUFA,R0       ;READ THE CHAR OUT OF THE FIFO.
4432 023766  016701  156322              MOV     RXINTC,R1       ;GET THE INTERUPT COUNT.
4433 023772  005201                      INC     R1              ;INCREMENT THE COUNT.
4434 023774  001402                      BEQ     2$              ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
4435 023776  010167  156312              MOV     R1,RXINTC       ;SAVE NEW COUNT VALUE.
4436 024002  016701  156310      2$:     MOV     RXINTF,R1       ;GET THE RX INTERRUPT FLAGS.
4437 024006  052701  000001              BIS     #BIT0,R1        ;SET THE RX INTERRUPT HAS OCCURRED FLAG.
4438 024012  032767  000001  156310      BIT     #BIT0,TXINTF    ;TEST THE "TX INT HAS OCCURRED" FLAG.
4439 024020  001402                      BEQ     4$              ;SKIP SETTING ERROR FLAG IF NO TX INT.
4440 024022  052701  040000              BIS     #BIT14,R1       ;SET THE INTERRUPT ORDER ERROR FLAG.
4441 024026  010167  156264      4$:     MOV     R1,RXINTF       ;UPDATE THE RX INTERRUPT FLAGS.
4442 024032                      60$:    PASS                    ;RESTORE GPRS.
     024032  004736                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
4443 024034  000002              RTI
```

```
                        .SBTTL  INTERUPT SERVICE ROUTINE        - RXINPT -
                        ;** ************************************************************************
                        ;*      - RECEIVE CHARACTER INPUT INTERRUPT SERVICE ROUTINE -
                        ;*      THIS SERVICE ROUTINE INPUTS A CHARACTER FROM THE DUT AND LOADS THE
                        ;*      CHAR (COMPLETE WITH STATUS FLAGS) INTO A RECEIVE CHAR BUFFER IN
                        ;*      MEMORY.  THE INTERRUPT IS ALSO COUNTED.  THE RECEIVE CHAR BUFFER IS
                        ;*      MONITORED TO ENSURE THAT IT DOES NOT OVERFLOW.
                        ;*
                        ;* INPUTS:      BUFEND - LABELS THE END OF THE HOST MEMORY BUFFER.
                        ;*              BUFPTR - CONTAINS ADDRESS OF NEXT FREE BUFFER LOCATION.
                        ;*              CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
                        ;*              RBUFA - CONTAINS THE ADDRESS OF THE RBUF DUT REGISTER.
                        ;*              RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERUPTS.
                        ;*              RxINTF - RX INTERRUPT FLAGS.
                        ;*
                        ;* OUTPUTS:     BUFPTR - CONTAINS UPDATED ADDRESS OF NEXT FREE BUFFER LOCATION.
                        ;*              RXINTC - CONTAINS THE UPDATED INTERUPT COUNT.
                        ;*              RXINTF - RX INT FLAGS (BIT 15 SET IF RX.DATA.AVAIL IS CLEAR).
                        ;*
                        ;* CALLING SEQUENCE:    PUT THE ADDRESS OF THE LABEL RXINPT IN THE VECTOR
                        ;*                      LOCATION.
                        ;*
                        ;* COMMENTS:    IN CASE OF OVERFLOW OF THE MEMORY BUFFER, BUFPTR WILL BE
                        ;*              MAINTAINED EQUAL TO BUFEND AND THE WORD AT BUFPTR 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.
                        ;-- ************************************************************************

024036                  RXINPT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024036  004567  157734                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
024042  017701  156174          MOV     @CSRA,R1                ;READ THE CONTENTS OF THE CSR.
024046  032701  000200          BIT     #BIT7,R1                ;TEST RX.DATA.AVAIL BIT.
024052  001003                  BNE     2$                      ;BRANCH AROUND SETTING FLAG IF BIT IS SET.
024054  052767  100000  156234  BIS     #BIT15,RXINTF           ;SET THE RX.DATA.AVAIL CLEAR FLAG.
024062  016701  156226  2$:     MOV     RXINTC,R1               ;GET THE INTERUPT COUNT.
024066  005201                  INC     R1                      ;INCREMENT THE COUNT.
024070  001402                  BEQ     4$                      ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
024072  010167  156216          MOV     R1,RXINTC               ;SAVE NEW COUNT VALUE.
024076  016702  156200  4$:     MOV     BUFPTR,R2               ;GET THE POINTER TO NEXT FREE BUFFER WORD.
024102  017722  156136          MOV     @RBUFA,(R2)+            ;READ A CHAR FROM THE FIFO INTO BUFFER.
024106  020267  157614          CMP     R2,BUFEND               ;TEST FOR POINTER BEYOND END OF BUFFER.
024112  103002                  BHIS    60$                     ;SKIP THE PTR UPDATE IF PTR OUT OF BOUNDS.
024114  010267  156162          MOV     R2,BUFPTR               ;UPDATE THE BUFFER POINTER.
024120                  60$:    PASS                            ;RESTORE GPRS.
024120  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
024122  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.
                                ;*************************************************************************

024124  021627  017764        TP4RTN:: CMP     (SP),@ADRPTR    ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
024130  001402                        BEQ     2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
024132  000177  156164                JMP     @TP4VEC         ;IF NOT,JUMP TO NORMAL 004 TRAP SERVICE RTN.
024136  052767  100000  156154  2$:   BIS     #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
024144  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
                        ;** ************************************************************************

4546 024146             TXINTR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024146  004567  157624                     JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
4547 024152  016701  156150            MOV     TXINTC,R1       ;GET THE TX INTERUPT COUNT.
4548 024156  005201                    INC     R1              ;INCREMENT THE COUNT.
4549 024160  102001                    BVC     2$              ;BRANCH IF NO OVERFLOW OCCURRED.
4550 024162  005301                    DEC     R1              ;RESET THE COUNT TO 177777.
4551 024164  010167  156136    2$:     MOV     R1,TXINTC       ;SAVE NEW COUNT VALUE.
4552 024170  016703  156134            MOV     TXINTF,R3       ;GET THE TX INTERRUPT FLAGS.
4553 024174  017702  156042            MOV     @CSRA,R2        ;READ THE CSR.
4554 024200  100402                    BMI     4$              ;SKIP SETTING OF FLAG IF TX.ACTION IS SET.
4555 024202  052703  100000            BIS     #BIT15,R3       ;SET THE TX.ACTION CLEAR FLAG.
4556 024206  052703  000001    4$:     BIS     #BIT0,R3        ;SET THE TX INT HAS OCCURRED FLAG.
4557 024212  010367  156112            MOV     R3,TXINTF       ;UPDATE THE TX INTERRUPT FLAGS.
4558 024216             60$:    PASS                            ;RESTORE GPRS.
     024216  004736                             JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
4559 024220  000002            RTI
```

```
4568
4569
4570                                    .SBTTL  REPORT CODING SECTION
4571
4572                                    ;++
4573                                    ; THE REPORT CODING SECTION CONTAINS THE
4574                                    ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
4575                                    ;--
4576
4577 024222                                    BGNRPT
      024222                                                                L$RPT::
4578
4579 024222                                    EXIT    RPT
      024222  000167                                                              .WORD   U$JMP
      024224  000000                                                              .WORD   L10017-2 ,
4580
4581                                    .EVEN
4582
4583 024226                                    ENDRPT
      024226                                                                L10017::
      024226  104425                                                              TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE

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

024230                                  BGNPROT
024230                                                                  L$PROT::

024230  177777                          -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
024232  177777                          -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
024234  177777                          -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

024236                                  ENDPROT
```

```
                                .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.
                                ;*
                                ;*****************************************************************************
                                ;--
024236                                  BGNINIT
024236                                                                  L$INIT::
                                ;SEE IF PROGRAM JUST STARTED, BR IF YES
                                        READEF  #EF.START
024236  012700  000040                                                  MOV     #EF.START,R0
024242  104447                                                          TRAP    C$REFG
024244                                  BCOMPLETE       NEWSTA
024244  103416                                                          BCS     NEWSTA
                                ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
                                        READEF  #EF.RESTART
024246  012700  000037                                                  MOV     #EF.RESTART,R0
024252  104447                                                          TRAP    C$REFG
024254                                  BCOMPLETE       NEWRES
024254  103556                                                          BCS     NEWRES
                                ;SEE IF THIS IS A NEW PASS, BR IF YES
                                        READEF  #EF.NEW
024256  012700  000035                                                  MOV     #EF.NEW,R0
024262  104447                                                          TRAP    C$REFG
024264                                  BCOMPLETE       NEWPAS
024264  103555                                                          BCS     NEWPAS
                                ;SEE IF PROGRAM WAS JUST CONTINUED
                                        READEF  #EF.CONTINUE
024266  012700  000036                                                  MOV     #EF.CONTINUE,R0
024272  104447                                                          TRAP    C$REFG
024274                                  BNCOMPLETE      GETPRM
024274  103161                                                          BCC     GETPRM
024276  000167  000540                  JMP     ENDIT
024302                          NEWSTA:
024302                                  BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
024302  104433                                                          TRAP    C$RESET
                                ;++
                                ; SET UP FOR LINE TIME CLOCK INTERRUPTS.
                                ;-
024304                                  CLOCK   L,R1            ;GET THE CLOCK PARAMETERS.
024304  012700  000114                                                  MOV     #L,R0
024310  104462                                                          TRAP    C$CLCK
024312  010001                                                          MOV     R0,R1
024314  012167  156014                  MOV     (R1)+,CLKCSR            ;STORE CLOCK CSR ADDRESS.
024320  012167  156012                  MOV     (R1)+,CLKBRL            ;STORE CLOCK BUS REQ INT LEVEL.
024324  012167  156010                  MOV     (R1)+,CLKVEC            ;STORE CLOCK INTERRUPT VECTOR.
024330  012167  156006                  MOV     (R1)+,CLKHRZ            ;STORE CLOCK FREQUENCY.
024334  026727  156002  000062          CMP     CLKHRZ,#50.             ;TEST FOR 50HZ LINE FREQUENCY.
024342  001004                          BNE     2$                      ;BRANCH IF CLOCK IS NOT 50HZ.
```

INITIALIZE SECTION

```
4669 024344  012767  000024  156002          MOV     #20.,MSTICK             ;INDICATE 20MS PER CLOCK TICK.
4670 024352  000403                          BR      4$
4671 024354  012767  000021  155772  2$:     MOV     #17.,MSTICK             ;INDICATE 17 MS PER CLOCK TICK.
4672 024362                          4$:     SETVEC  CLKVEC,#CLKINT,#PRIO6   ;INITIALIZE CLOCK INTERRUPT VECTOR.
     024362  012746  000300                                                          MOV     #PRIO6,-(SP)
     024366  012746  023706                                                          MOV     #CLKINT,-(SP)
     024372  016746  155742                                                          MOV     CLKVEC,-(SP)
     024376  012746  000003                                                          MOV     #3,-(SP)
     024402  104437                                                                  TRAP    C$SVEC
     024404  062706  000010                                                          ADD     #10,SP
4673 024410  016700  155726                  MOV     CLKHRZ,R0               ;INITIALIZE THE BREAK COUNT
4674 024414  006300                          ASL     R0                      ; TO CAUSE A BREAK
4675 024416  010067  155730                  MOV     R0,BCOUNT               ; EVERY 2 SECONDS.
4676 024422                                  SETPRI  #PRIO5                  ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
     024422  012700  000240                                                          MOV     #PRIO5,R0
     024426  104441                                                                  TRAP    C$SPRI
4677                                 ;+
4678                                 ; ENABLE THE LINE TIME CLOCK (LTC) CHECKING TO MAKE SURE THAT THE CSR
4679                                 ; IS ACCESSABLE.
4680                                 ;    FIRST SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
4681                                 ;-
4682 024430  016767  153350  155664          MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
4683 024436  012767  024124  153340          MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4684                                 ;+
4685                                 ;    ENABLE LTC CHECKING FOR 004 TRAP IN CASE CSR IS NOT THERE.
4686                                 ;-
4687 024444  005067  155650                  CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
4688 024450  012767  000100  155654          MOV     #BIT6,WORD1     ;SET UP TO SET BIT6 OF THE LTC CSR.
4689 024456  012700  002332                  MOV     #WORD1,R0       ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
4690 024462  016701  155646                  MOV     CLKCSR,R1       ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
4691 024466  004767  173260                  JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
4692 024472  016767  155624  153304          MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4693 024500  103403                          BCS     6$              ;IF NO TRAP, LTC IS THERE SO CONTINUE.
4694 024502  005067  155634                  CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
4695 024506  000402                          BR      8$              ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
4696                                 ;+
4697                                 ; CALIBRATE THE DELAY ROUTINE MILLI-SECOND DELAY COUNT VALUE.
4698                                 ;-
4699 024510  004767  173012          6$:     JSR     PC,CALMSL
4700                                 ;+
4701                                 ; CHECK FOR MEMMORY MANAGEMENT PRESENT ON THIS MACHINE.
4702                                 ; IF MEM MGT IS PRESENT, DISABLE IT.
4703                                 ;-
4704 024514  016767  153264  155600  8$:     MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
4705 024522  012767  024124  153254          MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4706 024530  005067  155564                  CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
4707 024534  005067  155572                  CLR     WORD1           ;PREPARE TO CLEAR THE MEM MGT SR0 REGISTER.
4708 024540  012700  002332                  MOV     #WORD1,R0       ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
4709 024544  016701  155610                  MOV     MMSR0,R1        ;SELECT MEM MGT SR0 REGISTER AS DESTINATION.
4710 024550  005067  155606                  CLR     MMPRES          ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
4711 024554  005067  155604                  CLR     MMENAB          ;INDICATE MEM MGT IS NOT ENABLED.
4712 024560  004767  173166                  JSR     PC,CKTRAP       ;CLEAR THE MEM MGT SR0 REG AND CHECK FOR TRAP.
4713 024564  016767  155532  153212          MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4714 024572  103003                          BCC     10$             ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
4715 024574  012767  000001  155560          MOV     #1,MMPRES       ;INDICATE THAT MEM MGT IS PRESENT.
4716 024602  005067  155504          10$:    CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4717 024606  000167  000006                  JMP     NEWPAS          ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
```

```
                                NEWRES: BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
                                                                                        TRAP    C$RESET
                                        CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
                                NEWPAS:
                                        MOV     #-1,UNITN       ;RESET LOGICAL DEVICE TO -1
                                ;+
                                ; INCREMENT THE PASS COUNTER, CORRECT FOR ANY OVERFLOW.
                                ;  THIS COUNTER IS USED IN THE ROM VERSION TEST.
                                ;-
                                         INC    PASCNT          ;INCREMENT THE PASS COUNTER.
                                        BNE     GETPRM          ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
                                         DEC    PASCNT          ;SET PASS COUNT TO 177777 OCTAL.

                                ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
                                GETPRM:
                                        INC     UNITN           ;INCREMENT LOGICAL DEVICE NUMBER
                                        CMP     UNITN,L$UNIT    ;SEE IF MAXIMUM UNIT NO. EXCEEDED
                                        BGE     NEWPAS          ;BR IF YES

                                        GPHARD  UNITN,R1        ;GET P-TABLE POINTER INTO R1
                                                                                        MOV     UNITN,R0
                                                                                        TRAP    C$GPHRD
                                                                                        MOV     R0,R1
                                        BCOMPLETE       30$     ;BR IF DEVICE AVAILABLE
                                                                                        BCS     30$
                                        BR      GETPRM          ;SKIP THIS DEVICE


                                        ;************ HARDWARE PARAMETER MOVING CODE ****************
                                30$:    MOV     (R1)+,CSRA      ;STORE DHU-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
                                        MOV     (R1)+,R2        ;GET THE RX INTERRUPT VECTOR ADDRESS.
                                        MOV     R2,RXVECA       ;STORE RX INT VECTOR ADDRESS.
                                        ADD     #4,R2           ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
                                        MOV     R2,TXVECA       ;STORE TX INT VECTOR ADDRESS.
                                        MOV     (R1)+,ACTLNS    ;STORE DHU-11 ACTIVE LINE BIT MAP
                                        MOVB    (R1),BRLEVL     ;STORE DHU-11 INTERUPT BUS REQUEST LEVEL
                                ;+
                                ;       CALCULTATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
                                ;       DEVICE REGISTER ADDRESS TABLE.
                                ;-
                                        MOV     CSRA,R1         ;COPY CSR ADDRESS
                                        INC     R1              ;INCREMENT CSR ADDRESS
                                        INC     R1              ; COPY BY 2.
                                        MOV     #7,R3           ;SET UP REGISTER COUNT
                                        MOV     #RBUFA,R2       ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
                                12$:    MOV     R1,(R2)+        ;STORE REGISTER ADDRESS IN TABLE
                                        INC     R1              ;INCREMENT REGISTER ADDRESS
                                        INC     R1              ; BY 2,FOR THE NEXT DEVICE REGISTER.
                                        DEC     R3              ;DECREMENT REGISTER COUNT
                                        BNE     12$             ;LOOP IF NOT DONE

                                ;+
                                ; INITIALISE THE BMP CODE QUEUE.
                                ;-
                                         MOV    #BMPCQB,R0      ;GET THE START ADDRESS OF THE QUEUE.
                                         MOV    #BMPCQE,R1      ;GET THE END ADDRESS OF THE QUEUE.
```

INITIALIZE SECTION

```
4770 024764  010067  155534            MOV     R0,BMPCQP       ;SET THE POINTER TO THE START OF THE QUEUE.
4771 024770  005020              14$:   CLR     (R0)+           ;CLEAR OUT THE CONTENTS OF THE QUEUE.
4772 024772  020001                     CMP     R0,R1           ;CHECK IF END OF QUEUE HAS BEEN REACHED.
4773 024774  103775                     BLO     14$             ;LOOP IF NOT ALL DONE.
4774                                ;+
4775                                ; REPORT THE UNIT NUMBER IF THE SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
4776                                ;  AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
4777                                ;-
4778 024776  032767  000020  155220      BIT     #BIT4,OPTION    ;CHECK IF THE QUESTION WAS ANSWERED YES.
4779 025004  001416                     BEQ     16$             ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
4780 025006  026727  155000  000001      CMP     L$UNIT,#1       ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
4781 025014  003412                     BLE     16$             ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
4782 025016                              PRINTF  #MFUNIT,UNITN   ;REPORT UNIT NUMBER.
     025016  016746  155214                                                    MOV     UNITN,-(SP)
     025022  012746  005433                                                    MOV     #MFUNIT,-(SP)
     025026  012746  000002                                                    MOV     #2,-(SP)
     025032  010600                                                            MOV     SP,R0
     025034  104417                                                            TRAP    C$PNTF
     025036  062706  000006                                                    ADD     #6,SP
4783 025042                        16$:
4784
4785 025042  005067  155240        ENDIT:  CLR     CTRLCF          ;CLR THE CTRL-C TEST ABORT FLAG.
4786                                ;+
4787                                ; SET THE PROCESSOR PRIORITY TO DISABLE ALL INTERRUPTS.
4788                                ;-
4789 025046                              SETPRI  #PRIO7                  ;SET PROCESSOR PRIORITY TO 7.
     025046  012700  000340                                                    MOV     #PRIO7,R0
     025052  104441                                                            TRAP    C$SPRI
4790 025054                        ENDINIT
     025054                                                             L10021:
     025054  104411                                                            TRAP    C$INIT
4791
4792            000000              TNUM = 0                ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.
```

```
4801
4802
4803                                    .SBTTL  AUTODROP SECTION
4804
4805
4806                                    ;++
4807                                    ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
4808                                    ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
4809                                    ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
4810                                    ; DROPPED FROM TESTING.
4811                                    ;--
4812
4813 025056                                     BGNAUTO
      025056                                                                L$AUTO::
4814
4821
4822 025056                                     ENDAUTO
      025056                                                                L10022:
      025056  104461                                                               TRAP    C$AUTO
```

```
4831
4832
4833                                    .SBTTL  CLEANUP CODING SECTION
4834
4835                                    ;++
4836                                    ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4837                                    ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
4838                                    ;--
4839
4840 025060                                    BGNCLN
     025060                                                                          L$CLEAN::
4841
4850 025060  005767  155222                     TST     CTRLCF          ;DID WE GET HERE BY CTRL-C FROM TEST?
4851 025064  001401                             BEQ     2$              ;CTRL-C FROM TEST?  NO, SKIP BUS RESET.
4852 025066                                     BRESET                  ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
     025066  104433                                                                    TRAP    C$RESET
4853 025070                                    2$:
4854 025070                                     EXIT    CLN
     025070  104432                                                                    TRAP    C$EXIT
     025072  000002                                                                    .WORD   L10023-.
4855
4867
4868                                            .EVEN
4869
4870 025074                                    ENDCLN
     025074                                                                  L10023:
     025074  104412                                                                    TRAP    C$CLEAN
```

```
                                .SBTTL  DROP UNIT SECTION

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

                                        BGNDU
                                                                                L$DU::
                                         PRINTF #DROP,R0        ;REPORT UNIT THAT HAS BEEN DROPPED.
025076  010046                                                                 MOV     R0,-(SP)
025100  012746  025122                                                         MOV     #DROP,-(SP)
025104  012746  000002                                                         MOV     #2,-(SP)
025110  010600                                                                 MOV     SP,R0
025112  104417                                                                 TRAP    C$PNTF
025114  062706  000006                                                         ADD     #6,SP
025120  000427                          BR      EDROP         ;BRANCH AROUND THE MESSAGE.

025122  045     101     040   DROP:     .ASCIZ/%A UNIT%D6%A DROPPED FROM FURTHER TESTING.%N/
025125  125     116     111
025130  124     045     104
025133  066     045     101
025136  040     104     122
025141  117     120     120
025144  105     104     040
025147  106     122     117
025152  115     040     106
025155  125     122     124
025160  110     105     122
025163  040     124     105
025166  123     124     111
025171  116     107     056
025174  045     116     000
                                .EVEN
                        EDROP:
                                EXIT    DU
025200  000167                                                                 .WORD   J$JMP
025202  000000                                                                 .WORD   L10024-2-.

                                ENDDU
                                                                        L10024:
025204  104453                                                                 TRAP    C$DU
```

```


                                .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.
                                ;--

025206                                  BGNAU
025206                                                                                  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.
                                ;**********************************************************************************

025206                                  EXIT    AU
025206  000167                                                                                  .WORD   J$JMP
025210  000000                                                                                  .WORD   L10025-2-.


                                        .EVEN

025212                                  ENDAU
025212                                                                                  L10025:
025212  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.
                                ;*
                                ;*****************************************************************************
                                ;--

025214                                  BGNTST
025214                                                                  T1::
                        000001          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025214  012767  000001  155102           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (1)
025222  012767  177777  155056           MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
025230  012767  000145  156532           MOV    #101.,ERRNBR    ;SET THE TEST ERROR NUMBER IN THE TABLE.
025236  012767  005560  156526           MOV    #EM0103,ERRMSG  ;SET UP THE TEST FAILURE MESSAGE IN THE TABLE.
025244  012767  015510  156522           MOV    #ER0101,ERRBLK  ;SET-UP THE ERROR ROUTINE IN THE ERROR TABLE.
                                ;+
                                ; SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
                                ;-
025252  016767  152526  155042           MOV    4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
025260  012767  024124  152516           MOV    #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
025266  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.
                                ;-
025270  016700  154746                   MOV    CSRA,R0         ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
025274  012701  025466                   MOV    #52$,R1         ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
025300  004767  172446                   JSR    PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
025304  103402                           BCS    4$              ;IF NO TRAP, BYPASS ERROR.
025306  052705  100001                   BIS    #100001,R5      ;SET FATAL READ ERROR FLAGS.
025312  042767  000017  000146  4$:      BIC    #17,52$         ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
025320  010100                           MOV    R1,R0           ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
025322  016701  154714                   MOV    CSRA,R1         ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
025326  004767  172420                   JSR    PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
025332  103403                           BCS    6$              ;IF NO TRAP, BYPASS ERROR.
025334  052705  100002                   BIS    #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
025340  000434                           BR     40$             ;EXIT AND REPORT FATAL ERROR.
                                ;+
                                ;    NOW, WE TEST EACH REGISTER FOR THIS LINE.
                                ;-
025342  012702  000010           6$:     MOV    #8.,R2          ;INIT REGISTER COUNTER TO 8.
025346  016767  154670  000110           MOV    CSRA,50$        ;INITIALIZE THE REGISTER POINTER.
025354  016700  000104           8$:     MOV    50$,R0          ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
025360  012701  025466                   MOV    #52$,R1         ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
025364  004767  172362                   JSR    PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
025370  103402                           BCS    10$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
025372  052705  100001                   BIS    #100001,R5      ;SET FATAL READ ERROR FLAGS.
025376  010100                   10$:     MOV    R1,R0           ;USE OLD DEST AS SRC FOR CKTRAP MOVE
025400  016701  000060                   MOV    50$,R1          ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.
```

```
 4999 025404  004767  172342                JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
 5000 025410  103402                        BCS     12$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
 5001 025412  052705  100002                 BIS     #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
 5002 025416  005267  000042          12$:    INC     50$             ;INCREMENT THE REGISTER
 5003 025422  005267  000036                 INC     50$             ; POINTER BY 2.
 5004 025426  005302                        DEC     R2              ;COUNT THE REGISTER.
 5005 025430  001351                        BNE     8$              ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
 5006
 5007                                ;+
 5008                                ; DONE CHECKING DEVICE REGISTER ADDRESSES.
 5009                                ;    REPORT ANY ERRORS AND EXIT.
 5010                                ;-
 5011
 5012 025432  016767  154664  152344  40$:     MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
 5013 025440  005705                        TST     R5              ;CHECK THE ERROR FLAGS.
 5014 025442  100012                        BPL     60$             ;EXIT ROUTINE IF NO ERRORS.
 5015                                ;+
 5016                                ; REPORT "DEVICE REGISTER ACCESS TEST FAILED"
 5017                                ;-
 5018 025444                                ERROR
      025444  104460                                                                TRAP    C$ERROR
 5019
 5020
 5021 025446                                DODU    UNITN           ;DROP THIS UNIT FROM FUTHER TESTING.
      025446  016700  154564                                                        MOV     UNITN,R0
      025452  104451                                                                TRAP    C$DODU
 5022 025454  005067  154626                 CLR     CTRLCF          ;INDICATE NO CTRL-C ABORT FROM TEST.
 5023 025460                                DOCLN                   ;ABORT THIS SUB PASS.
      025460  104444                                                                TRAP    C$DCLN
 5024 025462  000402                        BR      60$             ;
 5025
 5026                                ;***************** LOCAL STORAGE. ***************************************
 5027 025464  000000                 50$:    .WORD   0               ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 5028 025466  000000                 52$:    .WORD   0               ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
 5029                                ;******************** END ***************************************************
 5030
 5031 025470  005067  154612          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 5032 025474                                ENDTST
      025474                                                                    L10026:
      025474  104401                                                                TRAP    C$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.
                                        ;*
                                        ;-- *****************************************************************
 025476                                         BGNTST
      025476                                                                             T2::
              000002                            TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 025476 012767  000002  154620                  MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (2)
 025504 012767  177777  154574                  MOV     # 1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 025512                                         SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
      025512  012700  000240                                                                    MOV     #PRI05,R0
      025516  104441                                                                            TRAP    C$SPRI
 025520 012767  000001  156240                  MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 025526 012767  005616  156236                  MOV     #EM0201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
 025534 012767  016042  156232                  MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                        ;+
                                        ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                        ;-
 025542 012701  011610                          MOV     #5000.,R1       ;TIME OUT VALUE IS 5.0 SECONDS.
 025546 012702  000040                          MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
 025552 005003                                  CLR     R3              ;WAITING FOR BIT TO CLEAR.
 025554 016704  154462                          MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
 025560 004767  172400                          JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
 025564 103410                                  BCS     2$              ;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.
                                        ;-
 025566                                         BRESET                  ;NO, TRY TO JOG DEVICE WITH BUS RESET.
      025566  104433                                                                            TRAP    C$RESET
 025570 004767  174550                          JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
 025574 012701  011610                          MOV     #5000.,R1       ;TIME OUT VALUE IS 5.0 SECONDS.
 025600 004767  172360                          JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
 025604 103016                                  BCC     4$              ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
                                        ;+
                                        ; SET THE MASTER RESET BIT AND VERIFY THAT IT CLEARS WITHIN THE PROPER TIME.
                                        ;-
 025606 012701  011610                  2$:     MOV     #5000.,R1       ;TIME OUT VALUE IS 5.0 SECONDS.
 025612 010214                                  MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
 025614 004767  172344                          JSR     PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
 025620 103010                                  BCC     4$              ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
 025622 012702  011610                          MOV     #5000.,R2
 025626 160102                                  SUB     R1,R2           ;CALCULATE # OF MS FOR MR TO CLEAR.
 025630 001413                                  BEQ     6$              ;GO REPORT ERROR IF MR CLEAR IMMEDIATELY
 025632 020227  000764                          CMP     R2,#500.
 025636 002417                                  BLT     8$              ;GO REPORT ERROR IF MR CLEAR IN < 1/2 SECOND.
 025640 000424                                  BR      60$             ;EXIT THE TEST WITHOUT ERROR.
                                        ;+
                                        ; ERROR REPORTS:
                                        ;
                                                ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
 025642 012767  000311  156120          4$:     MOV     #201.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
 025650 012701  005651                          MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
 025654                                         ERROR                   ;REPORT ERROR.          >>>>> ERROR 0201 <<<<<
```

```
                025654  104460                                                          TRAP    C$ERROR
5087            025656  000415                  BR      60$             ;EXIT THE TEST.
5088
5089                                    ;REPORT MR BIT CLEAR IMMEDIATELY AFTER DUT RESET.
5090            025660  012767  000312  156102  6$:     MOV     #202.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
5091            025666  012701  006024                  MOV     #EM0203,R1      ;SELECT ERROR MESSAGE.
5092            025672                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR 0202 <<<<<
                025672  104460                                                          TRAP    C$ERROR
5093            025674  000406                  BR      60$             ;EXIT THE TEST.
5094
5095                                    ;REPORT MR CLEAR WITHIN 1/2 SECOND OF DUT RESET.
5096            025676  012767  000313  156064  8$:     MOV     #203.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
5097            025704  012701  006167                  MOV     #EM0204,R1      ;SELECT ERROR MESSAGE.
5098            025710                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR 0203 <<<<<
                025710  104460                                                          TRAP    C$ERROR
5099
5100            025712                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
                025712  012700  000340                                                  MOV     #PRI07,R0
                025716  104441                                                          TRAP    C$SPRI
5101            025720  005067  154362                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5102            025724                                  ENDTST
                025724                                                          L10027:
                025724  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.
                                ;*
                                ;-- ************************************************************************
025726                          BGNTST
025726                                                                          T3::
                 000003         TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025726 012767  000003  154370    MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (3)
025734 012767  177777  154344    MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
025742                          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
025742 012700  000240                                                   MOV     #PRI05,R0
025746 104441                                                           TRAP    C$SPRI
025750 012767  000001  156010    MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
025756 012767  006346  156006    MOV    #EM0301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
025764 012767  016042  156002    MOV    #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;*
                                ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;-
025772 012701  011610            MOV    #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
025776 012702  000040            MOV    #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
026002 005003                    CLR    R3              ;WAITING FOR BIT TO CLEAR.
026004 016704  154232            MOV    CSRA,R4         ;BIT IS IN THE DUT'S CSR.
026010 004767  172150           JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026014 103410                    BCS    2$              ;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.
                                ;-
026016                          BRESET                  ;NO, TRY TO JOG DEVICE WITH BUS RESET.
026016 104433                                                           TRAP    C$RESET
026020 004767  174320           JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
026024 012701  011610            MOV    #5000.,R1       ;TIME OUT VALUE IS 5.0 SECONDS.
026030 004767  172130           JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026034 103024                    BCC    6$              ;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.
                                ;-
026036 012701  000310   2$:      MOV    #200.,R1        ;TIME-OUT VALUE IS 1/5 SECOND.
026042 010214                    MOV    R2,(R4)         ;SET THE DUT MASTER RESET BIT.
026044 004767  174274           JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
026050 004767  172110           JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026054 103007                   BCC     4$              ;GO FIND OUT WHAT IS WRONG IF MR NOT CLEAR.
026056 012702  000310            MOV    #200.,R2
026062 160102                    SUB    R1,R2           ;CALCULATE # OF MS FOR MR TO CLEAR.
026064 020227  000012            CMP    R2,#10.
026070 002415                   BLT     8$              ;GO REPORT ERROR IF MR CLEAR IN < 10 MS.
026072 000431                    BR     60$             ;EXIT THE TEST WITHOUT ERROR.
                                ;*
                                ; MR DID NOT CLEAR WITHIN 1/5 SECOND, SEE IF IT CLEARS WITHIN 5 SECONDS.
                                ;-
026074 012701  011300   4$:      MOV    #4800.,R1       ;TIME OUT VALUE IS 5 SECONDS MINUS 1/5 SECOND.
026100 004767  172060           JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026104 103416                    BCS    10$             ;GO REPORT ERROR IF MR CLEARED FINALLY.
```

```
                                        ;*
                                        ; ERROR REPORTS:
                                        ;-
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
026106  012767  000455  155654  6$:       MOV     #0301.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026114  012701  005651                    MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
026120                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #0301 <<<<<
026120  104460                                                                                    TRAP    C$ERROR
026122  000415                            BR      60$             ;EXIT THE TEST.

                                    ;REPORT MR BIT CLEAR WITHIN 10 MS AFTER DUT RESET.
026124  012767  000456  155636  8$:       MOV     #0302.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026132  012701  006420                    MOV     #EM0302,R1      ;SELECT ERROR MESSAGE.
026136                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #0302 <<<<<
026136  104460                                                                                    TRAP    C$ERROR
026140  000406                            BR      60$             ;EXIT THE TEST.

                                    ;REPORT MR CLEARED BETWEEN 1/5 SECOND AND 5 SECONDS OF DUT RESET.
026142  012767  000457  155620  10$:      MOV     #0303.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026150  012701  006560                    MOV     #EM0303,R1      ;SELECT ERROR MESSAGE.
026154                                    ERROR                   ;REPORT ERROR.          >>>>> ERROR #0303 <<<<<
026154  104460                                                                                    TRAP    C$ERROR

026156                            60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
026156  012700  000340                                                                          MOV     #PRI07,R0
026162  104441                                                                                  TRAP    C$SPRI
026164  005067  154116                    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
026170                                    ENDTST
026170                                                                            L10030:
026170  104401                                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RXCHRA -
                                ;++ ************************************************************************
                                ;*              - RBUF REGISTER RX CHARACTER FIELD TEST -
                                ;*      THIS TEST VERIFIES THAT THE RX CHARACTER FIELD OF THE DUT RBUF REGISTER
                                ;*      APPEARS TO BE FUNCTIONING CORRECTLY.  THIS TEST USES THE CODES WHICH
                                ;*      SHOULD BE IN THE FIFO AFTER A BOARD RESET AND SKIP SELFTEST SEQUENCE.
                                ;*
                                ;-- ************************************************************************
                                        BGNTST
                                                                                T4::
                                        SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
026172  012700  000240                                                          MOV     #PRIO5,R0
026176  104441                                                                 TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026200  012767  000004  154116          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (4)
026206  012767  177777  154072          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
026214  012767  000001  155544          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026222  012767  006737  155542          MOV     #EM0401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026230  012767  016042  155536          MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
026236  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
026242  012702  000040                  MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
026246  005003                          CLR     R3              ;WAITING FOR BIT TO CLEAR.
026250  016704  153766                  MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
026254  010214                          MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
026256  004767  174062                  JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
026262  004767  171676                  JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026266  103015                          BCC     4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;+
                                ; READ 6 CHARACTERS FROM THE DUT AND VERIFY THAT THEY ARE VALID SELFTEST
                                ;  CODES.
                                ;-
026270  012400                          MOV     (R4)+,R0        ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
026272  012701  000006                  MOV     #6,R1           ;INITIALIZE THE LOOP COUNTER.
026276  011402                  2$:     MOV     (R4),R2         ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
026300  010200                          MOV     R2,R0
026302  042700  177476                  BIC     #177476,R0      ;REMOVE ALL BUT BITS SPECIFIC TO SELFTEST CODE.
026306  020027  000201                  CMP     R0,#201         ;CHECK THAT BITS 0,6, AND 7 ARE CORRECT.
026312  001012                          BNE     6$              ;GO REPORT ERROR IF CODE IS NOT SELFTEST CODE.
026314  005301                          DEC     R1              ;COUNT THIS LOOP ITERATION.
026316  001367                          BNE     2$              ;LOOP IF NOT ALL LINES DONE.
026320  000415                          BR      60$             ;EXIT TEST, NO ERROR FOUND.

                                ;+
                                ; ERROR REPORTS:
                                ;-
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
026322  012767  000621  155440  4$:      MOV     #0401.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026330  012701  005651                  MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
026334                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0401 <<<<<
026334  104460                                                                          TRAP    C$ERROR
026336  000406                          BR      60$             ;EXIT THE TEST.

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

```
5234 026340  012767  000622  155422  6$:     MOV     #0402.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5235 026346  012701  007015                  MOV     #EM0402,R1      ;SELECT ERROR MESSAGE.
5236 026352                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0402 <<<<<
     026352  104460                                                                              TRAP    C$ERROR
5237
5238 026354                          60$:    SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
     026354  012700  000340                                                                      MOV     #PRIO7,R0
     026360  104441                                                                              TRAP    C$SPRI
5239 026362  005067  153720                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5240 026366                                  ENDTST
     026366                                                                      L10031:
     026366  104401                                                                              TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - RXFFDA -
                                ;++ ************************************************************************
                                ;*              - RBUF REGISTER RX FLAG FIELD TEST -
                                ;*      THIS TEST VERIFIES THAT THE FIELD OF 3 FLAG BITS IN THE RBUF READS
                                ;*      AS ALL ONES WHEN THE SELFTEST CODES ARE BEING READ FROM THE DUT
                                ;*      AFTER A BOARD RESET AND SKIP SELFTEST SEQUENCE.
                                ;*
                                ;-- ************************************************************************
026370                                  BGNTST
026370                                                                          T5::
026370                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
026370  012700  000240                                                        MOV     #PRI05,R0
026374  104441                                                                TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026376  012767  000005  153720          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (5)
026404  012767  177777  153674          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
026412  012767  000001  155346          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026420  012767  007165  155344          MOV     #EM0501,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026426  012767  016042  155340          MOV     #ER0201,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
026434  012701  011610                  MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
026440  012702  000040                  MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
026444  005003                          CLR     R3              ;WAITING FOR BIT TO CLEAR.
026446  016704  153570                  MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
026452  010214                          MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
026454  004767  173664                  JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
026460  004767  171500                  JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026464  103013                          BCC     4$              ;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.
                                ;-
026466  012400                          MOV     (R4)+,R0        ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
026470  012701  000010                  MOV     #8.,R1          ;INITIALIZE THE LOOP COUNTER.
026474  011402                  2$:     MOV     (R4),R2         ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
026476  012700  070000                  MOV     #70000,R0
026502  040200                          BIC     R2,R0           ;CALCULATE BIT MAP OF CLEAR RX ERROR FLAGS.
026504  001012                          BNE     6$              ;GO REPORT ERROR IF NOT ALL RX ERROR FLAGS SET.
026506  005301                          DEC     R1              ;COUNT THIS LOOP ITERATION.
026510  001371                          BNE     2$              ;LOOP IF NOT ALL LINES DONE.
026512  000415                          BR      60$             ;EXIT TEST, NO ERROR FOUND.
                                ;+
                                ; ERROR REPORTS:
                                ;
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
026514  012767  000765  155246  4$:      MOV     #0501.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
026522  012701  005651                  MOV     #EM0202,R1      ;SELECT ERROR MESSAGE.
026526                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0501 <<<<<
026526  104460                                                                        TRAP    C$ERROR
026530  000406                          BR      60$             ;EXIT THE TEST.
                                    ;REPORT ONE OR MORE RX ERROR FLAGS FOUND CLEAR WITH SELFTEST CODE.
026532  012767  000766  155230  6$:      MOV     #0502.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
```

```
5295 026540  012701  007241                MOV     #EM0502,R1      ;SELECT ERROR MESSAGE.
5296 026544                                ERROR                   ;REPORT ERROR.          >>>>> ERROR #0502 <<<<<
     026544  104460                                                                            TRAP    C$ERROR
5297
5298 026546                          60$:  SETPRI  #PRI07           ;DISABLE ALL INTERRUPTS.
     026546  012700  000340                                                                    MOV     #PRI07,R0
     026552  104441                                                                            TRAP    C$SPRI
5299 026554  005067  153526                CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5300 026560                                ENDTST
     026560                                                                       L10032:
     026560  104401                                                                            TRAP    C$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.
                                ;*
                                ;-- ******************************************************************************
026562                                  BGNTST
026562                                                                          T6::
026562                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
026562  012700  000240                                                                  MOV     #PRIO5,R0
026566  104441                                                                        TRAP    C$SPRI
        000006                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
026570  012767  000006  153526           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (6)
026576  012767  177777  153502           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
026604  012767  000001  155154           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
026612  012767  007575  155152           MOV     #EMO601,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
026620  012767  016042  155146           MOV     #ERO201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;*
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
026626  012701  011610                   MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
026632  012702  000040                   MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
026636  005003                           CLR     R3             ;WAITING FOR BIT TO CLEAR.
026640  016704  153376                   MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
026644  010214                           MOV     R2,(R4)        ;SET THE DUT MASTER RESET BIT.
026646  004767  173472                   JSR     PC,SKPSTS      ;SKIP THE SELFTEST.
026652  004767  171306                   JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
026656  103016                           BCC     4$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                ;*
                                ; CHECK THAT THE RX DATA AVAILABLE BIT IS SET.
                                ;-
026660  032714  000200                   BIT     #BIT7,(R4)     ;TEST THE DUT RX.DATA.AVAIL BIT.
026664  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.
                                ;-
026666  012705  001130                   MOV     #600.,R5       ;ALLOW READING 600 CHARS BEFORE ERROR.
026672  010403                           MOV     R4,R3
026674  012300                           MOV     (R3)+,R0       ;CALCULATE THE RBUF ADDRESS.
026676  011300                2$:        MOV     (R3),R0        ;READ A CHARACTER FROM THE RX FIFO.
026700  032714  000200                   BIT     #BIT7,(R4)     ;TEST THE DUT RX.DATA.AVAIL BIT.
026704  001427                           BEQ     60$            ;EXIT TEST WITHOUT ERROR IF RX.DATA.AVAIL CLR.
026706  005305                           DEC     R5             ;COUNT THE CHARACTER JUST READ.
026710  001372                           BNE     2$             ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
026712  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.
026714  012767  001131  155046  4$:       MOV     #0601.,ERRNBR  ;SET THE ERROR NUMBER IN ERROR TABLE.
026722  012701  005651                    MOV     #EMO202,R1     ;SELECT ERROR MESSAGE.
026726                                   ERROR                  ;REPORT ERROR.          >>>>> ERROR #0601 <<<<<
026726  104460                                                                        TRAP    C$ERROR
```

```
5355 026730  000415                          BR      60$             ;EXIT THE TEST.
5356
5357                                    ;REPORT THAT RX.DATA.AVAIL BIT WAS NOT SET AFTER A RESET COMPLETION.
5358 026732  012767  001132  155030  6$:     MOV     #0602.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5359 026740  012701  007637                  MOV     #EM0602,R1      ;SELECT ERROR MESSAGE.
5360 026744                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0602 <<<<<
     026744  104460                                                                              TRAP    C$ERROR
5361 026746  000406                          BR      60$             ;EXIT THE TEST.
5362
5363                                    ;REPORT THAT RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO.
5364 026750  012767  001133  155012  8$:     MOV     #0603.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5365 026756  012701  010017                  MOV     #EM0603,R1      ;SELECT ERROR MESSAGE.
5366 026762                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR #0603 <<<<<
     026762  104460                                                                              TRAP    C$ERROR
5367
5368 026764                          60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     026764  012700  000340                                                                      MOV     #PRI07,R0
     026770  104441                                                                              TRAP    C$SPRI
5369 026772  005067  153310                   CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5370 026776                                  ENDTST
     026776                                                                          L10033:
     026776  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.
                                        ;*
                                        ;-- ************************************************************************
027000                                          BGNTST
027000                                                                                  T7::
027000                                          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
027000  012700  000240                                                                  MOV     #PRI05,R0
027004  104441                                                                        TRAP    C$SPRI
        000007                           TNUM == TNUM + 1               ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027006  012767  000007  153310                  MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (7)
027014  012767  177777  153264                  MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
027022  012767  000001  154736                  MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027030  012767  010202  154734                  MOV     #EMO701,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
027036  012767  016042  154730                  MOV     #ERO201,ERRSUB  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                        ;+
                                        ; SET THE DUT C.R MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                        ;  AND WAIT UP  C 5 SECONDS FOR THE MR BIT TO CLEAR.
                                        ;-
027044  012701  011610                          MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
027050  012702  000040                          MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
027054  005003                                  CLR     R3              ;WAITING FOR BIT TO CLEAR.
027056  016704  153160                          MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
027062  010214                                  MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
027064  004767  173254                        JSR     PC,SKPSTS       ;SKIP THE SELFTEST.
027070  004767  171070                        JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
027074  103012                                BCC     4$              ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                        ;+
                                        ; CHECK THAT THE RX DATA VALID BIT IS SET.
                                        ;-
027076  012400                                  MOV     (R4)+,R0        ;INCREMENT POINTER TO PNT TO DUT RBUF REG.
027100  005714                                  TST     (R4)            ;TEST THE DUT RX.DATA.VALID BIT.
027102  100016                                BPL     6$              ;GO REPORT ERROR IF BIT IS NOT SET.
                                        ;+
                                        ; READ CHARACTERS FROM THE DUT RX FIFO AND WAIT FOR RX.DATA.VALID TO GO CLEAR.
                                        ;-
027104  012705  001130                          MOV     #600.,R5        ;ALLOW READING 600 CHARS BEFORE ERROR.
027110  011400                        2$:       MOV     (R4),R0         ;READ A CHARACTER FROM THE RX FIFO.
027112  100027                                BPL     60$             ;EXIT TEST WITHOUT ERROR IF BIT IS CLEAR.
027114  005305                                  DEC     R5              ;COUNT THE CHARACTER JUST READ.
027116  001374                                BNE     2$              ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
027120  000416                                  BR      8$              ;GO REPORT ERROR IF RX.DATA.VALID WOULDN'T CLR.
                                        ;+
                                        ; ERROR REPORTS:
                                        ;-
                                            ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
027122  012767  001275  154640        4$:       MOV     #0701.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
027130  012701  005651                          MOV     #EMO202,R1      ;SELECT ERROR MESSAGE.
027134                                          ERROR                   ;REPORT ERROR.          >>>>> ERROR #0701 <<<<<
027134  104460                                                                                TRAP    C$ERROR
027136  000415                                  BR      60$             ;EXIT THE TEST.
```

```
5425                                        ;REPORT THAT RX.DATA.VALID BIT WAS NOT SET AFTER A RESET COMPLETION.
5426 027140  012767  001276  154622  6$:     MOV     #0702.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5427 027146  012701  010245                  MOV     #EM0702,R1      ;SELECT ERROR MESSAGE.
5428 027152                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0702 <<<<<
     027152  104460                                                                              TRAP    C$ERROR
5429 027154  000406                         BR      60$             ;EXIT THE TEST.
5430
5431                                        ;REPORT THAT RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO.
5432 027156  012767  001277  154604  8$:     MOV     #0703.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5433 027164  012701  010425                  MOV     #EM0703,R1      ;SELECT ERROR MESSAGE.
5434 027170                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR #0703 <<<<<
     027170  104460                                                                              TRAP    C$ERROR
5435
5436 027172                          60$:   SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     027172  012700  000340                                                                      MOV     #PRI07,R0
     027176  104441                                                                              TRAP    C$SPRI
5437 027200  005067  153102                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5438 027204                                 ENDTST
     027204                                                                         L10034:
     027204  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.
                                ;*
                                ;-- ******************************************************************************
027206                                  BGNTST
027206                                                                          T8::
027206                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
027206  012700  000240                                                          MOV     #PRIO5,R0
027212  104441                                                                  TRAP    C$SPRI
        000010                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027214  012767  000010  153102           MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (8)
027222  012767  177777  153056           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
027230  012767  000001  154530           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027236  012767  010610  154526           MOV     #EM0801,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
                                ;+
                                ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
                                ;  AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
027244  012701  011610                   MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
027250  012702  000040                   MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
027254  005003                           CLR     R3             ;WAITING FOR BIT TO CLEAR.
027256  016704  152760                   MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
027262  010214                           MOV     R2,(R4)        ;SET THE DUT MASTER RESET BIT.
027264  004767  173054                   JSR     PC,SKPSTS      ;SKIP THE SELFTEST.
027270  004767  170670                   JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
027274  103016                           BCC     4$             ;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.
                                ;-
027276  005001                           CLR     R1             ;CLEAR THE LINE COUNTER.
027300  012400                           MOV     (R4)+,R0       ;INCREMENT POINTER TO PNT TO THE DUT RBUF REG.
027302  011402                  2$:      MOV     (R4),R2        ;READ A CHARACTER FROM THE DUT RX FIFO.
027304  010203                           MOV     R2,R3
027306  000303                           SWAB    R3
027310  042703  177760                   BIC     #177760,R3     ;REMOVE ALL BUT LINE NUMBER BITS.
027314  020301                           CMP     R3,R1          ;COMPARE WITH EXPECTED LINE NUMBER.
027316  001017                           BNE     6$             ;GO REPORT ERROR IF LINE NUMBERS DON'T MATCH.
027320  005201                           INC     R1             ;INCREMENT THE EXPECTED LINE NUMBER.
027322  020127  000010                   CMP     R1,#8.         ;COMPARE NUMBER OF CODES READ WITH MAX.
027326  001365                           BNE     2$             ;LOOP UNTIL CODES FOR ALL LINES ARE READ.
027330  000423                           BR      60$            ;EXIT TEST WITHOUT ERROR.

                                ;+
                                ; ERROR REPORTS:
                                ;-
                                    ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
027332  012767  001441  154430  4$:       MOV     #0801.,ERRNBR  ;SET THE ERROR NUMBER IN ERROR TABLE.
027340  012767  016136  154426           MOV     #ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
027346  012701  005651                   MOV     #EM0202,R1     ;SELECT ERROR MESSAGE.
027352                                   ERROR                  ;REPORT ERROR.          >>>>> ERROR #0801 <<<<<
027352  104460                                                                  TRAP    C$ERROR
```

```
5493 027354  000411                          BR      60$             ;EXIT THE TEST.
5494
5495                                  ;REPORT THAT RX LINE NUMBER FIELD IS WRONG FOR SELFTEST CODE.
5496 027356  012767  001442  154404  6$:     MOV     #0802.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5497 027364  012767  016136  154402          MOV     #ERR0C3,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5498 027372  012701  010656                  MOV     #EM0802,R1      ;SELECT ERROR MESSAGE.
5499 027376                                  ERROR                   ;REPORT ERROR.          >>>>> ERROR 0802 <<<<<
     027376  104460                                                                          TRAP    C$ERROR
5500
5501 027400                          60$:    SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
     027400  012700  000340                                                                  MOV     #PRIO7,R0
     027404  104441                                                                          TRAP    C$SPRI
5502 027406  005067  152674                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5503 027412                                  ENDTST
     027412                                                                      L10035:
     027412  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.
                                ;*
                                ;-- ****************************************************************************
027414                                  BGNTST
027414                                                                  T9::
027414                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
027414  012700  000240                                                          MOV     #PRIO5,R0
027420  104441                                                                  TRAP    C$SPRI
000011                           TNUM ** TNUM + 1               ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
027422  012767  000011  152674            MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.                (9)
027430  012767  177777  152650            MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
027436  012767  000001  154322            MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
027444  012767  001605  154316            MOV     #0901.,ERRNBR   ;SET THE ERROR NUMBER.
027452  012767  010726  154312            MOV     #EM0901,ERRMSG  ;SET THE ERROR MESSAGE
                                ;*
                                ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                ;-
027460  012701  005670                    MOV     #3000.,R1       ;TIME OUT VALUE IS 3.0 SECONDS.
027464  012702  000040                    MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
027470  005003                            CLR     R3              ;WAITING FOR BIT TO CLEAR.
027472  016704  152544                    MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
027476  004767  170462                    JSR     PC,MSLGET       ;WAIT FOR DUT CSR.MR BIT TO CLEAR.
027502  103027                            BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
                                ;*
                                ; RESET THE DUT, SKIP THE SELF-TEST.
                                ;-
027504  010214                            MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
027506  004767  172632                    JSR     PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
                                ;*
                                ; WAIT FOR MASTER RESET TO CLEAR, DELAY FOR 500 MILLI-SECS BEFORE PURGING
                                ; THE FIFO.
                                ;-
027512  012704  000764                    MOV     #500.,R4        ;TIME OUT VALUE IS 500 MILLI SECONDS.
027516  004767  170402                    JSR     PC,DELAY        ;WAIT FOR BMP TO BEGIN EXECUTION.
027522  004767  171014                    JSR     PC,PURFIFO      ;PURGE THE FIFO, SAVING ANY BMP CODES.
027526  103015                            BCC     50$             ;ABORT THE TEST IF THE FIFO DID NOT CLEAR.
                                ;*
                                ; REPORT THE ERROR IF ANY BMP CODES WERE FOUND.
                                ;-
027530  016702  152770                    MOV     BMPCQP,R2       ;GET THE CONTENTS OF THE POINTER TO THE BMP Q.
027534  012703  002526                    MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
027540  020203                            CMP     R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
027542  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
```

```
 5559 027544  012701  010766                MOV     #EM0902,R1      ;PASS THE MESSAGE TO BE REPORTED.
 5560 027550                                ERRDF   0901,EM0901,ER9301 ;              >>>>> ERROR #0901 <<<<<.
      027550  104455                                                                   TRAP    C$ERDF
      027552  001605                                                                   .WORD   901
      027554  010726                                                                   .WORD   EM0901
      027556  017262                                                                   .WORD   ER9301
 5561 027560  000405                           BR      60$
 5562
 5563 027562  012767  001606  154200  50$:    MOV     #902.,ERRNBR    ;SET >>>>> ERROR #0902 <<<<<.
 5564 027570  004767  172676                JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
 5565
 5566 027574                        60$:    SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
      027574  012700  000340                                                           MOV     #PRIO7,R0
      027600  104441                                                                   TRAP    C$SPRI
 5567 027602  005067  152500                CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5568 027606                                ENDTST
      027606                                                                  L10036:
      027606  104401                                                                   TRAP    C$ETST
```

```
5570
5571
5572                                    .SBTTL  HARDWARE TEST           - SKSELF -
5573                                    ;** ************************************************************************
5574                                    ;*              - SKIP SELF-TEST TEST -
5575                                    ;*      THIS TEST VERIFIES THAT THE DUT SKIPS THE SELF-TEST WITHIN THE
5576                                    ;*      TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS
5577                                    ;*      COMPLETION.
5578                                    ;*
5579                                    ;-- ************************************************************************
5580 027610                                        BGNTST
     027610                                                                                          T10::
5581 027610                                        SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
     027610  012700  000240                                                                         MOV     #PRIO5,R0
     027614  104441                                                                                 TRAP    C$SPRI
5582            000012                              TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5583 027616  012767  000012  152500                MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (10)
5584 027624  012767  177777  152454                MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
5585 027632  012767  000001  154126                MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5586 027640  012767  011022  154124                MOV     #EM1001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5587 027646  012767  016136  154120                MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5588                                    ;*
5589                                    ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5590                                    ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5591                                    ;-
5592 027654  012701  011610                        MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
5593 027660  012702  000040                        MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
5594 027664  005003                                CLR     R3              ;WAITING FOR BIT TO CLEAR.
5595 027666  016704  152350                        MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
5596 027672  004767  170266                        JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5597 027676  103037                                BCC     50$             ;ABORT THE TEST IF MR DID NOT CLEAR.
5598                                    ;*
5599                                    ; DETERMINE IF THE DUT TAKES TOO SHORT OR TOO LONG A TIME TO SKIP THE SELF-TEST
5600                                    ;  SET-UP A TIME-OUT OF 50 MILLI-SECOND, IF MR IS CLEAR IN LESS THAN 10 MILLI
5601                                    ;    -SECOND, OR GREATER THAN 50 MILLI-SECONDS, REPORT THE ERROR.
5602                                    ;-
5603 027700  012701  000062                        MOV     #50.,R1         ;TIME-OUT VALUE IS 50 MILLI-SECONDS.
5604 027704  010214                                MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
5605 027706  004767  172432                        JSR     PC,SKPSTS       ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
5606 027712  004767  170246                        JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5607 027716  103011                                BCC     2$              ;GO REPORT ERR IF SKIPPING STEST TOOK TOO LONG.
5608 027720  020127  000050                        CMP     R1,#40.
5609 027724  003015                                BGT     4$              ;GO REP ERR IF SELFTEST COMPLETED IN < 10 MS.
5610                                    ;*
5611                                    ; SELF-TEST COMPLETED WITHIN 10 MILLI-SEC TO 50 MILLI-SECONDS.
5612                                    ;  VERIFY THAT THE SELF-TEST CODES IN THE FIFO ARE "GOOD" CODES ,IF THE DUT
5613                                    ;  SUCCESSFULLY COMPLETED THE SELF-TEST.
5614                                    ; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 1003 THRU 1007 <<<<<.
5615                                    ;-
5616 027726  012767  001753  154034                MOV     #1003.,ERRNBR   ;SET ERROR NUMBER TO 1003.
5617 027734  004767  171554                        JSR     PC,RSTRPT       ;CHECK SELF-TEST CODES IN THE FIFO.
5618 027740  000423                                BR      60$             ;EXIT TEST.
5619                                    ;*
5620                                    ; ERROR REPORTS:
5621                                    ;-
5622                                       ;REPORT SKIP SELF-TEST TOOK TOO LONG.
5623 027742  012767  001751  154020  2$:           MOV     #1001.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
```

```
5624 027750  012701  011055                MOV     #EM1002,R1      ;SELECT ERROR MESSAGE.
5625 027754                                ERROR                   ;REPORT ERROR.          >>>>> ERROR #1001 <<<<<
     027754  104460                                                                            TRAP    C$ERROR
5626 027756  000414                        BR      60$             ;EXIT THE TEST.
5627
5628                                  ;REPORT SKIP SELF-TEST COMPLETED TOO SOON.
5629 027760  012767  001752  154002  4$:   MOV     #1002.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5630 027766  012701  011142                MOV     #EM1003,R1      ;SELECT ERROR MESSAGE.
5631 027772                                ERROR                   ;REPORT ERROR.          >>>>> ERROR #1002 <<<<<
     027772  104460                                                                            TRAP    C$ERROR
5632 027774  000405                        BR      60$             ;EXIT THE TEST.
5633
5634 027776  012767  001753  153764  50$:  MOV     #1003.,ERRNBR   ;SET ERROR NUMBER.
5635 030004  004767  172462                JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
5636
5637 030010                          60$:  SETPRI  #PRIO7           ;DISABLE ALL INTERRUPTS.
     030010  012700  000340                                                                    MOV     #PRIO7,R0
     030014  104441                                                                            TRAP    C$SPRI
5638 030016  005067  152264                CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5639 030022                                ENDTST
     030022                                                                          L10037:
     030022  104401                                                                            TRAP    C$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.
                                        ;*
                                        ;-- ************************************************************************
030024                                          BGNTST
030024                                                                                  T11::
030024                                          SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
030024  012700  000240                                                                  MOV     #PRIO5,R0
030030  104441                                                                        TRAP    C$SPRI
                000013                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
030032  012767  000013  152264                   MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (11)
030040  012767  177777  152240                   MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
030046  012767  000001  153712                   MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
030054  012767  011220  153710                   MOV     #EM1101,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
030062  012767  016136  153704                   MOV     #ERO503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                        ;*
                                        ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                        ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                        ;-
030070  012701  011610                           MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
030074  012702  000040                           MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
030100  005003                                   CLR     R3             ;WAITING FOR BIT TO CLEAR.
030102  016704  152134                           MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
030106  004767  170052                           JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030112  103044                                   BCC     50$            ;ABORT THE TEST IF MR DID NOT CLEAR.
                                        ;*
                                        ; RESET THE DUT, SKIP THE SELF-TEST.
                                        ;-
030114  010214                                   MOV     R2,(R4)        ;SET THE DUT MASTER RESET BIT.
030116  004767  172222                           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.
                                        ;-
030122  012701  000005                           MOV     #5,R1          ;TIME-OUT VALUE IS 5 MILLI-SECONDS.
030126  012702  020000                           MOV     #BIT13,R2      ;WAITING FOR DIAGNOSTIC FAIL BIT.
030132  010203                                   MOV     R2,R3          ;WAITING FOR BIT TO SET.
030134  016704  152102                           MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
030140  004767  170020                           JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
030144  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.
                                        ;
030146  012701  000017                           MOV     #15.,R1        ;TIME-OUT VALUE IS 15 MILLI-SECONDS.
030152  005003                         2$:       CLR     R3             ;WAITING FOR BIT TO CLEAR.
030154  004767  170004                           JSR     PC,MSLGET      ;WAIT FOR DUT CSR_DF BIT TO CLEAR.
030160  103012                                   BCC     4$             ;IF DIAG FAIL DID NOT CLEAR, GO REPORT ERROR.
030162  010105                                   MOV     R1,R5          ;SAVE THE REMAINING TIME-OUT VALUE.
```

```
 5695 030164  012701  000001                MOV     #1,R1           ;SET TIME-OUT OF 1 MILLI-SECOND.
 5696 030170  052703  020000                BIS     #BIT13,R3       ;WAIT FOR BIT TO SET.
 5697 030174  004767  167764                JSR     PC,MSLGET       ;DOUBLE CHECK TO ELIMINATE NOISE PROBLEMS.
 5698 030200  103016                        BCC     60$             ;EXIT IF DIAG_FAIL BIT STILL CLEAR.
 5699 030202  010501                        MOV     R5,R1           ;PASS THE REMAINING TIME-OUT VALUE.
 5700 030204  000762                        BR      2$              ;LOOP TO CHECK AGAIN.
 5701                                ;+
 5702                                ; ERROR REPORTS:
 5703                                ;-
 5704                                    ;REPORT DIAGNOSTIC FAIL BIT BAD.
 5705 030206  012767  002115  153554  4$:   MOV     #1101.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
 5706 030214  012701  011517                MOV     #EM1205,R1      ;SELECT ERROR MESSAGE.
 5707 030220                                ERROR                   ;REPORT ERROR.          >>>>> ERROR #1101 <<<<<
      030220  104460                                                                            TRAP    C$ERROR
 5708 030222  000405                        BR      60$             ;EXIT THE TEST.
 5709
 5710 030224  012767  002116  153536  50$:  MOV     #1102.,ERRNBR   ;SET THE ERROR NUMBER FOR TSABRT RTN.
 5711 030232  004767  172234                JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
 5712
 5713 030236                        60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      030236  012700  000340                                                                    MOV     #PRI07,R0
      030242  104441                                                                            TRAP    C$SPRI
 5714 030244  005067  152036                CLR     CTRLCF          ;INDICATE THAT WE COMPLETED A TEST.
 5715 030250                                ENDTST
      030250                                                                        L10040:
      030250  104401                                                                           TRAP    C$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.
                                ;*
                                ;-- ************************************************************************
030252                          BGNTST
030252                                                                  T12::
030252                          SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
030252  012700  000240                                                  MOV     #PRIO5,R0
030256  104441                                                         TRAP    C$SPRI
                                 TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
030260  012767  000014  152036   MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (12)
030266  012767  177777  152012   MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
030274  012767  000001  153464   MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
030302  012767  011273  153462   MOV     #EM1201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
030310  012767  016136  153456   MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                ;-
030316  012701  011610           MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
030322  012702  000040           MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
030326  005003                   CLR     R3              ;WAITING FOR BIT TO CLEAR.
030330  016704  151706           MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
030334  004767  167624           JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030340  103067                   BCC     50$             ;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.
                                ;-
030342  012701  011610           MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
030346  010214                   MOV     R2,(R4)         ;SET THE DUT MASTER RESET BIT.
030350  004767  167610           JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030354  103034                   BCC     4$              ;GO REPORT ERROR SELFTEST TOOK TOO LONG.
030356  012702  011610           MOV     #5000.,R2
030362  160102                   SUB     R1,R2           ;CALCULATE # OF MS SELFTEST TO COMPLETE.
030364  020227  000062           CMP     R2,#50.
030370  002435                   BLT     6$              ;SELFTEST SKIPPED?  YES, GO REPORT ERROR.
030372  020227  000764           CMP     R2,#500.
030376  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.
                                ;-
030400  032714  020000           BIT     #BIT13,(R4)     ;DETERMINE IF THE DIAG_FAIL BIT IS CLEAR.
030404  001412                   BEQ     2$              ;SKIP ERROR REPORT IF BIT IS CLEAR.
                               ;REPORT DIAGNOSTIC FAIL BIT BAD.
030406  012767  002264  153354   MOV     #1204.,ERRNBR   ;SET ERROR NUMBER TO IN ERROR TABLE.
030414  012701  011517           MOV     #EM1205,R1      ;SELECT THE ERROR MESSAGE.
030420                           ERROR                   ;               >>>>> ERROR #1204 <<<<<
030420  104460                                                           TRAP    C$ERROR
                                ;+
                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
```

```
5770                                    ;-
5771 030422  032767  000100  151574          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
5772 030430  001440                        BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5773                                                               ;DURING THE SOFTWARE QUESTIONS.
5774                                    ;+
5775                                    ; VERIFY THAT THE SELF-TEST CODES IN THE FIFO ARE "GOOD" CODES ,IE THE DUT
5776                                    ; SUCCESSFULLY COMPLETED THE SELF-TEST.
5777                                    ; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 1205 THRU 1209 <<<<<.
5778                                    ;-
5779 030432  012767  002265  153330  2$:     MOV     #1205.,ERRNBR   ;SET ERROR NUMBER TO 1205.
5780 030440  004767  171050                JSR     PC,RSTRPT       ;CHECK SELF-TEST CODES IN THE FIFO.
5781 030444  000432                        BR      60$             ;EXIT TEST.
5782                                    ;+
5783                                    ; ERROR REPORTS:
5784                                    ;-
5785                                        ;REPORT SELF-TEST TOOK TOO LONG TO COMPLETE.
5786 030446  012767  002261  153314  4$:     MOV     #1201.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5787 030454  012701  011321                MOV     #EM1202,R1      ;SELECT ERROR MESSAGE.
5788 030460                                ERROR                   ;REPORT ERROR.          >>>>> ERROR #1201 <<<<<
     030460  104460                                                                            TRAP    C$ERROR
5789 030462  000423                        BR      60$             ;EXIT THE TEST.
5790
5791                                        ;REPORT SELF-TEST DID NOT EXECUTE AFTER DUT RESET.
5792 030464  012767  002262  153276  6$:     MOV     #1202.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5793 030472  012701  011463                MOV     #EM1204,R1      ;SELECT ERROR MESSAGE.
5794 030476                                ERROR                   ;REPORT ERROR.          >>>>> ERROR #1202 <<<<<
     030476  104460                                                                            TRAP    C$ERROR
5795 030500  000414                        BR      60$             ;EXIT THE TEST.
5796
5797                                    ;REPORT SELF-TEST COMPETED TOO SOON.
5798 030502  012767  002263  153260  8$:     MOV     #1203.,ERRNBR   ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5799 030510  012701  011405                MOV     #EM1203,R1      ;SELECT ERROR MESSAGE.
5800 030514                                ERROR                   ;REPORT ERROR.          >>>>> ERROR #1203 <<<<<
     030514  104460                                                                            TRAP    C$ERROR
5801 030516  000405                        BR      60$             ;EXIT THE TEST.
5802
5803 030520  012767  002272  153242  50$:    MOV     #1210.,ERRNBR   ;SET THE ERROR NUMBER FOR TSABRT RTN.
5804 030526  004767  171740                JSR     PC,TSABRT       ;REPORT NON-TEST RELATED ERROR.
5805
5806 030532                            60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     030532  012700  000340                                                                    MOV     #PRI07,R0
     030536  104441                                                                            TRAP    C$SPRI
5807 030540  005067  151542                    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5808 030544                                ENDTST
     030544                                                                             L10041:
     030544  104401                                                                            TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE TEST               - STFAIL -
                                        ;++ ********************************************************************
                                        ;*               - SELF TEST FAIL TEST -
                                        ;*       THIS TEST VERIFIES THAT THE DUT WILL REPORT SELFTEST ERRORS VIA THE
                                        ;*       FIFO.  AND THAT THE DIAGNOSTIC FAIL BIT WILL INDICATE THE ERROR.
                                        ;*       THIS IS ACCOMPLISHED VIA A SOFTWARE 'HOOK' IN THE SELF-TEST, WHICH
                                        ;*       FORCES A "PROC1 TO RAM ERROR" TO BE PLACED IN THE FIFO.
                                        ;*
                                        ;-- ********************************************************************
030546                                          BGNTST
     030546                                                                          T13::
030546                                          SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
     030546  012700  000240                                                                       MOV     #PRIO5,R0
     030552  104441                                                                               TRAP    C$SPRI
             000015                             TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
030554  012767  000015  151542                  MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.          (13)
030562  012767  177777  151516                  MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
030570  012767  000001  153170                  MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
030576  012767  011543  153166                  MOV     #EM1301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
030604  012767  016136  153162                  MOV     #ER0503,ERRBLK  ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
030612  012767  002425  153150                  MOV     #1301.,ERRNBR   ;SET ERROR NUMBER TO 1301.
                                        ;+
                                        ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                        ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                        ;-
030620  012701  011610                          MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
030624  012702  000040                          MOV     #BIT05,R2       ;WAITING FOR MASTER RESET BIT.
030630  005003                                  CLR     R3              ;WAITING FOR BIT TO CLEAR.
030632  016704  151404                          MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
030636  004767  167322                          JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030642  103064                                  BCC     50$             ;GO REPORT ERROR IF MR DID NOT CLEAR.

                                        ;+
                                        ; RESET THE DUT.  DELAY FOR 25 MILLI-SECONDS BEFORE WRITING THE FAIL_SELF_TEST
                                        ; CODE TO TBUFFCT REGISTER ON CHANNEL 0.
                                        ;-
030644  012777  000040  151370                  MOV     #BIT05,@CSRA    ;SET DUT MASTER RESET BIT, SELECT CHANNEL 0.
030652  012704  000031                          MOV     #25.,R4         ;PASS DELAY PERIOD OF 25 MILLI SECS.
030656  004767  167242                          JSR     PC,DELAY        ;WAIT FOR SELFTEST TO INITIALISE.
030662  012777  146314  151370                  MOV     #146314,@TXBFCA ;WRITE THE FAIL SELF-TEST CODE TO TBUFFCT REG.
                                        ;+
                                        ; WAIT UP TO 5 SECONDS FOR THE SELF-TEST TO COMPLETE.
                                        ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                        ;-
030670  005267  153074                          INC     ERRNBR          ;SET ERROR NUMBER TO 1302.
030674  012701  011610                          MOV     #5000.,R1       ;TIME-OUT VALUE IS 5.0 SECONDS.
030700  012702  000040                          MOV     #BIT05,R2       ;PASS THE BIT MAP OF THE BIT TO TEST.
030704  005003                                  CLR     R3              ;SET UP THE EXPECTED STATE.
030706  016704  151330                          MOV     CSRA,R4         ;BIT IS IN THE DUT'S CSR.
030712  004767  167246                          JSR     PC,MSLGET       ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
030716  103036                                  BCC     50$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
                                        ;+
                                        ; VERIFY THE DIAGNOSTIC FAIL BIT IS SET, INDICATING THE ERROR.
                                        ; REPORT ERROR IF DIAGNOSTIC FAIL BIT IS CLEAR.
                                        ;-
030720  005267  153044                          INC     ERRNBR          ;SET ERROR NUMBER TO 1303.
```

```
 5864 030724  032714  020000                 BIT     #BIT13,(R4)     ;CHECK THE STATE OF THE DIAG_FAIL BIT.
 5865 030730  001425                        BEQ     10$             ;GO REPORT ERROR IF DIAG_FAIL BIT CLEAR.
 5866                                ;+
 5867                                ; REMOVE THE 8 SELF-TEST CODES FORM THE FIFO, AND VERIFY THAT AT LEAST
 5868                                ; ONE IS A PROC1 TO RAM ERROR CODE (231).
 5869                                ;-
 5870 030732  005267  153032                 INC     ERRNBR          ;SET ERROR NUMBER TO 1304.
 5871 030736  012700  000010                 MOV     #8.,R0          ;SET MAXIMUM READ COUNT.
 5872 030742  005001                         CLR     R1              ;CLEAR THE CORRECT CODE COUNTER.
 5873 030744  016704  151274                 MOV     RBUFA,R4        ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
 5874 030750  011402                 6$:    MOV     (R4),R2         ;READ A CODE FROM THE FIFO.
 5875 030752  100020                        BPL     50$             ;GO REPORT ERROR IF THE FIFO IS EMPTY.
 5876 030754  042702  007400                 BIC     #7400,R2        ;REMOVE THE LINE NUMBER FROM THE CODE.
 5877 030760  120227  170231                 CMPB    R2,#170231      ;IS IT THE CORRECT ERROR CODE?.
 5878 030764  001001                        BNE     8$              ;SKIP NEXT INSTRUCTION, IF NOT A 231 CODE.
 5879 030766  005201                         INC     R1              ;INCREMENT COUNTER.
 5880 030770  005300                 8$:    DEC     R0              ;DECREMENT MAX READ COUNTER.
 5881 030772  001366                        BNE     6$              ;LOOP IF 8 CODES HAVE NOT BEEN READ.
 5882 030774  005701                         TST     R1              ;WERE ANY 231 CODES FOUND?.
 5883 030776  001010                        BNE     60$             ;YES, THEN EXIT.
 5884 031000  005267  152764                 INC     ERRNBR          ;NO, SET ERROR NUMBER TO 1305 AND REPORT ERROR.
 5885                            ;REPORT SELF-TEST ERROR REPORTING BAD.
 5886 031004  012701  011576         10$:    MOV     #EM1302,R1      ;SELECT ERROR MESSAGE.
 5887 031010                                 ERROR                   ;REPORT ERROR.          >>>>> ERROR <<<<<
      031010  104460                                                                              TRAP    C$ERROR
 5888 031012  000402                        BR      60$             ;EXIT THE TEST.
 5889
 5890 031014  004767  171452         50$:    JSR     PC,TSABRT       ;REPORT NON-RELATED TEST ERROR.
 5891
 5892 031020                         60$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      031020  012700  000340                                                                    MOV     #PRI07,R0
      031024  104441                                                                            TRAP    C$SPRI
 5893 031026  005067  151254                 CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 5894 031032                                 ENDTST
      031032                                                                        L10042:
      031032  104401                                                                            TRAP    C$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.
                                ;*
                                ;-- ******************************************************************************
031034                                  BGNTST
031034                                                                          T14::
031034                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
031034  012700  000240                                                        MOV     #PRIO5,R0
031040  104441                                                                TRAP    C$SPRI
        000016                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031042  012767  000016  151254          MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (14)
031050  012767  177777  151230          MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
031056  012767  000001  152702          MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
031064  012767  011635  152700          MOV     #EM1401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
031072  012767  016136  152674          MOV     #ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
                                ;+
                                ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
                                ;  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
                                ;-
031100  012701  005670                    MOV     #3000.,R1      ;TIME-OUT VALUE IS 3.0 SECONDS.
031104  012702  000040                    MOV     #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
031110  005003                            CLR     R3             ;WAITING FOR BIT TO CLEAR.
031112  016704  151124                    MOV     CSRA,R4        ;BIT IS IN THE DUT'S CSR.
031116  004767  167042                    JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
031122  103131                            BCC     50$            ;ABORT THE TEST IF MR DID NOT CLEAR.
                                ;+
                                ; SET THE MASTER RESET BIT, AND SKIP THE SELF TEST.
                                ;-
031124  010214                            MOV     R2,(R4)        ;SET THE MASTER RESET BIT.
031126  004767  171212                    JSR     PC,SKPSTS      ;SKIP THE SELF TEST.
031132  012701  011610                    MOV     #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
031136  004767  167022                    JSR     PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
031142  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.
                                ;-
031144  012705  000040                    MOV     #32.,R5        ;SET MAXIMUM TRY COUNTER.
031150  012703  00014?                    MOV     #99.,R3        ;SET AN INVALID ROM VERSION NUMBER FOR PROC 1.
031154  010304                            MOV     R3,R4          ;SET AN INVALID ROM VERSION NUMBER FOR PROC 2.
031156  012767  002574  152604          MOV     #1401.,ERRNBR  ;SET THE ERROR NUMBER TO 1401.
031164  012701  011674                    MOV     #EM1402,R1     ;SELECT MESSAGE TO BE REPORTED IF FIFO EMPTY.

031170  017702  151050          2$:       MOV     @RBUFA,R2      ;READ THE NEXT CHAR FROM THE FIFO.
031174  100077                            BPL     12$            ;GO REPORT ERROR IF FIFO EMPTY.
                                  ;+
                                  ; CHECK IF THE READ DATA IS A BMP CODE.
```

```
5950                                    ;-
5951 031176  012700  000301                     MOV     #301,R0         ;SET-UP A BIT MASK OF A BMP CODE.
5952 031202  040200                             BIC     R2,R0           ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
5953 031204  001003                           BNE     4$                ;BRANCH IF NOT A BMP CODE.
5954 031206  004767  171064                   JSR     PC,SAVBMP         ;SAVE THE BMP CODE ON THE QUEUE.
5955 031212  000435                           BR      8$                ;
5956                                    ;+
5957                                    ; CHECK IF THE READ DATA IS A SELF-TEST CODE.
5958                                    ;-
5959 031214  012700  000201            4$:      MOV     #201,R0         ;SET-UP A BIT MASK OF A SELFTEST CODE.
5960 031220  040200                             BIC     R2,R0           ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
5961 031222  001431                           BEQ     8$                ;BRANCH IF IT IS A SELFTEST CODE.
5962
5963                                    ;+
5964                                    ; THE READ DATA IS A ROM VERSION NUMBER, DETERMINE WHICH ONE IT IS.
5965                                    ;-
5966 031224  032702  000002                     BIT     #BIT1,R2        ;CHECK THE PROCESSOR NUMBER BIT IN THE CODE.
5967 031230  001407                           BEQ     6$                ;BRANCH IF IT IS PROC_1 ROM VERSION NUMBER.
5968 031232  010204                             MOV     R2,R4           ;SAVE PROC_2 ROM VERSION NUMBER.
5969 031234  042704  177603                     BIC     #177603,R4      ;CLEAR ANY UNWANTED BITS.
5970 031240  000241                             CLC                     ;CLEAR THE CARRY BIT.
5971 031242  006004                             ROR     R4              ;SHIFT THE CODES ALONG TO GET THE ROM
5972 031244  006004                             ROR     R4              ; VERSION NUMBER IN THE LOW 5 BITS.
5973 031246  000417                           BR      8$                ;
5974 031250  010203                    6$:      MOV     R2,R3           ;SAVE PROC_1 ROM VERSION NUMBER.
5975 031252  042703  177603                     BIC     #177603,R3      ;CLEAR ANY UNWANTED BITS.
5976 031256  000241                             CLC                     ;CLEAR THE CARRY BIT.
5977 031260  006003                             ROR     R3              ;SHIFT THE CODE ALONG TO GET THE ROM
5978 031262  006003                             ROR     R3              ; VERSION NUMBER IN THE LOW 5 BITS.
5979 031264  020427  000143                     CMP     R4,#99.         ;CHECK IF WE HAVE RECEIVE PROC_2 ROM CODE.
5980 031270  001016                           BNE     10$               ;GO REPORT BOTH ROM VERSION NUMBERS.
5981                                    ;+
5982                                    ; RECEIVED ROM VERSION NUMBERS OUT OF SEQUENCE.
5983                                    ;  IE, PROC_1'S ROM VERSION NUMBER FOUND IN THE FIFO BEFORE PROC_2'S.
5984                                    ;-
5985 031272  012701  011762                     MOV     #EM1403,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5986 031276  012767  002572  152464             MOV     #1402.,ERRNBR   ;SET THE ERROR NUMBER.
5987 031304  000433                           BR      12$               ;GO REPORT ERROR.
5988
5989 031306  005305                    8$:      DEC     R5              ;DECREMENT THE MAX TRY COUNTER.
5990 031310  001327                           BNE     2$                ;LOOP TO GET THE NEXT CHAR FROM THE FIFO.
5991 031312  012701  012035                     MOV     #EM1404,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5992 031316  012767  002573  152444             MOV     #1403.,ERRNBR   ;SET THE ERROR NUMBER.
5993 031324  000423                             BR      12$             ;GIVE UP, GO REPORT ERROR.
5994                                    ;+
5995                                    ; IF THIS IS THE FIRST PASS, AND SOFTWARE P-TABLE QUESTION WAS ANSWERED YES.
5996                                    ; THEN REPORT THE ROM VERSION NUMBERS TO THE OPERATOR.
5997                                    ;
5998 031326  032767  000001  150670    10$:      BIT     #BIT0,OPTION    ;CHECK ON THE STATE OF THE SOFTWARE SWITCH.
5999 031334  001431                           BEQ     60$               ;EXIT IF NO ROM VERSION PRINTOUT WAS REQUESTED.
6000 031336  026727  150750  000001             CMP     PASCNT,#1       ;CHECK IF THIS IS THE FIRST PASS.
6001 031344  003025                           BGT     60$               ;EXIT IF ROM VERS HAVE ALREADY BEEN REPORTED.
6002 031346                                     PRINTB #EF1401,R3,R4    ;PRINT THE ROM VERSION NUMBERS.
     031346  010446                                                                    MOV     R4,-(SP)
     031350  010346                                                                    MOV     R3,-(SP)
     031352  012746  004154                                                           MOV     #EF1401,-(SP)
     031356  012746  000003                                                           MOV     #3,-(SP)
```

```
        031362  010600                                                                  MOV     SP,R0
        031364  104414                                                                  TRAP    C$PNTB
        031366  062706  000010                                                          ADD     #10,SP
 6003   031372  000412                          BR      60$             ;EXIT THIS TEST.
 6004                                    ;
 6005                                    ; ERROR REPORTS:
 6006                                    ;
 6007   031374  012767  016254  152372  12$:    MOV     #ER1401,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
 6008   031402                          ERROR                           ;REPORT ERROR.          >>>>> ERROR <<<<<
        031402  104460                                                                  TRAP    C$ERROR
 6009   031404  000405                          BR      60$
 6010
 6011   031406  012767  002575  152354  50$:    MOV     #1405.,ERRNBR   ;SET UP ERROR NUMBER FOR TSABRT RTN.
 6012   031414  004767  171052                  JSR     PC,TSABRT       ;REPORT NON TEST RELATED ERROR.
 6013
 6014   031420                          60$:    SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
        031420  012700  000340                                                          MOV     #PRIO7,R0
        031424  104441                                                                  TRAP    C$SPRI
 6015   031426  005067  150654                  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 6016   031432                          ENDTST
        031432                                                          L10043:
        031432  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - CSRB4 -
                                ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;*                              CSR BIT 4 TEST -
                                ;*      THIS TEST VERIFIES THAT WHEN THIS BIT IS SET (AT THE SAME TIME
                                ;*      AS MASTER RESET) THE DUT REMAINS INACTIVE WITH THE MASTER RESET
                                ;*      BIT SET; AND WHEN CSR BIT 4 IS SUBSEQUENTLY CLEARED, THE BOARD
                                ;*      BECOMES ACTIVE AND REPORTS SIX SKIP SELFTEST CODES IN THE RXFIFO.
                                ;*      ANY BMP CODES FOUND IN THE FIFO ARE SAVED TO BE REPORTED LATER.
                                ;*
                                ;*---++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

031434                                  BGNTST
031434                                                                  T15::
031434                          SETPRI  @PRIO5          ;ALLOW LTC INTERRUPTS.
031434  012700  000240                                                  MOV     @PRIO5,R0
031440  104441                                                          TRAP    C$SPRI

                000017                   TNUM == TNUM + 1       ;INCREMENT THE ASSMBLY TIME TEST COUNTER.
031442  012767  000017  150654           MOV     @TNUM,TSTNUM   ;SET UP THE TEST NUMBER.
031450  012767  177777  150630           MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
031456  012767  000001  152302           MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
031464  012767  002735  152276           MOV     #1501.,ERRNBR  ;SET THE ERROR NUMBER TO 1501.
031472  012767  012155  152272           MOV     #EM1501,ERRMSG ;SET THE ERROR MESSAGE ADDR.
031500  012767  016136  152266           MOV     #ERC503,ERRBLK ;SET THE ERROR REPORTING ROUTINE.

                                ;+
                                ; WAIT FOR THE MASTER RESET BIT TO CLEAR, REPORT THE ERROR IF IT FAILS
                                ; TO CLEAR.
                                ;-
031506  012701  011610                  MOV     #5000.,R1       ;SET THE TIME-OUT VALUE OF 5 SEC.
031512  012702  000040                  MOV     #BIT05,R2       ;INDICATE TO TEST BIT 5.
031516  005003                          CLR     R3              ;INDICATE TO TEST FOR BIT CLEAR.
031520  016704  150516                  MOV     CSRA,R4         ;INDICATE TO TEST THE CSR REG.
031524  004767  166550                 JSR     PC,MSLOOP       ;WAIT FOR THE BIT TO CLEAR
031530  103115                        BCC     40$             ;JUMP TO REPORT ERROR IF BIT FAILED TO CLEAR.

                                ;+
                                ; SET THE MASTER RESET AND CSR BIT 4 BITS, AND THEN WAIT 5.5 SECS TO ENSURE
                                ; THAT THE MR BIT DOESN'T CLEAR.
                                ;-
031532  005267  152232                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 1502.
031536  012777  000060  150476          MOV     #60,@CSRA       ;RESET THE BOARD WITH BIT 4 SET.

                                ;+
                                ;VERIFY THAT CSR BIT 4 IS SET.
                                ;-
031544  017700  150472                  MOV     @CSRA,R0        ;READ THE CSR.
031550  032700  000020                  BIT     #BIT04,R0       ;TEST BIT 4.
031554  001477                        BEQ     40$             ;EXIT WITH ERROR IF THE BIT IS CLEAR.

                                ;+
                                ;WAIT 5 SECONDS FOR THE MR BIT TO CLEAR.
                                ;-
031556  005267  152206                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 1503.
031562  012701  011574                  MOV     #5500.,R1       ;SET THE TIME OUT VALUE OF 5.5 SECS.
031566  004767  166506                 JSR     PC,MSLOOP       ;WAIT FOR THE MASTER RESET BIT TO CLEAR.
031572  103470                        BCS     40$             ;REPORT THE ERROR IF THE MR BIT CLEARED.
                                ;+
```

```
                                        ; CLEAR CSR BIT 4 AND VERIFY THAT THE MASTER RESET BIT ALSO CLEARS.
                                        ;-
031574  017705  150442                          MOV     @CSRA,R5                ;READ THE CSR.
031600  042705  000020                          BIC     #BIT04,R5               ;CLEAR BIT 4.
031604  010577  150432                          MOV     R5,@CSRA                ;RESTORE THE CONTENTS OF THE CSR.
                                        ;+
                                        ;VERIFY THAT CSR BIT 4 CLEARED.
                                        ;-
031610  005267  152154                          INC     ERRNBR                  ;SET THE ERROR NUMBER TO 1504.
031614  017705  150422                          MOV     @CSRA,R5                ;READ THE CSR.
031620  032705  000020                          BIT     #BIT04,R5               ;TEST BIT 4.
031624  001053                                  BNE     40$                     ;BRANCH AND REPORT ERROR IF SET.
                                        ;+
                                        ;WAIT FOR THE MR BIT TO CLEAR.
                                        ;-
031626  005267  152136                          INC     ERRNBR                  ;SET THE ERROR NUMBER TO 1505.
031632  012701  000764                          MOV     #500.,R1                ;SET A TIME-OUT OF 1/2 SECS.
031636  004767  166436                          JSR     PC,MSLOOP               ;WAIT FOR THE MR BIT TO CLEAR.
031642  103044                                  BCC     40$                     ;JUMP AND REPORT ERROR,
                                                                                ; MR BIT FAILED TO CLEAR.
                                        ;+
                                        ; READ SIX CHARACTERS FROM THE RXFIFO AND VERIFY THEY ARE SKIP-SELFTEST
                                        ; CODES.  SAVE ANY BMP CODES FOUND TO BE REPORTED LATER.
                                        ;-
031644  012767  017126  152122                  MOV     #ER9008,ERRBLK          ;SET UP THE ERROR ROUTINE.
031652  012701  015241                          MOV     #EM9024,R1              ;SET THE ERROR MESSAGE,
                                                                                ; "IMPROPER SELFTEST CODE FOUND".
031656  012704  000006                          MOV     #6.,R4                  ;SET THE NUMBER OF CHAR'S TO READ.

031662  012767  002742  152100  2$:             MOV     #1506.,ERRNBR           ;SET THE ERROR NUMBER TO 1506.
031670  017702  150350                          MOV     @RBUFA,R2               ;READ A CODE FROM THE RXFIFO.
031674  100033                                  BPL     50$                     ;EXIT WITH ERROR IF THE FIFO IS EMPTY.
031676  010203                                  MOV     R2,R3                   ;COPY THE CODE.
031700  042703  177400                          BIC     #177400,R3              ;CLEAR THE LINE NUMBER AND ERROR FLAGS.
031704  012705  000301                          MOV     #301,R5                 ;SET THE BMP CODE MASK.
031710  040305                                  BIC     R3,R5                   ;CHECK IF THE CODE IS A BMP CODE.
031712  001003                                  BNE     4$                      ;AVOID SAVING THE BMP CODE IF IT ISN'T.
031714  004767  170356                          JSR     PC,SAVBMP               ;SAVE THE BMP CODE.
031720  000760                                  BR      2$                      ;AVOID COUNTING THIS CODE, AND BRANCH
                                                                                ;TO READ MORE DATA FROM THE RXFIFO.
031722  020327  000203          4$:             CMP     R3,#203                 ;IS THE CODE A SKIP SELFTEST ?
031726  001407                                  BEQ     6$                      ;BRANCH TO AVOID THE ERROR IF IT IS.
                                        ;+
                                        ; REPORT UNEXPECTED SELFTEST CODE FOUND.
                                        ;-
031730  005267  152034                          INC     ERRNBR                  ;SET THE ERROR NUMBER TO 1507.
031734                                          ERROR                           ;REPORT THE ERROR.
031734  104460                                                                                  TRAP    C$ERROR
                                        ;+
                                        ;IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED THEN EXIT
                                        ;THE TEST WITH THE TEST FAILURE MESSAGE.
                                        ;-
031736  032767  000100  150260                  BIT     #BIT06,OPTION           ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
031744  001411                                  BEQ     60$                     ;EXIT THE TEST IF IT HASN'T.

031746  005304                  6$:             DEC     R4                      ;DECREMENT THE CODE COUNT.
031750  001344                                  BNE     2$                      ;BRANCH AND READ ANOTHER CODE IF NOT ALL
```

```
6128
6129 031752  000406                    BR      60$             ;HAVE BEEN READ.
6130                                                            ;OTHERWISE, EXIT THE TEST, THE DUT HAS PASSED
6131                                                            ;THIS TEST.
6132
6133                            ;+
6134                            ; REPORT THE ERROR "CSR BIT 4 BAD" AND EXIT THIS TEST.
6135                            ;-
6136
6137 031754  012701  012203     40$:    MOV     #EM1502,R1      ;SET UP THE EXTENDED ERROR MESSAGE AS,
6138                                                            ; "CSR BIT 4 BAD".
6139 031760                             ERROR                   ;REPORT THE ERROR.
     031760  104460                                                         .RAP    C$ERROR
6140 031762  000402                     BR      60$             ;EXIT THE TEST.
6141
6142                            ;+
6143                            ; REPORT A NON-RELATED TEST ERROR.
6144 031764  004767  170502     50$:    JSR     PC,TSABRT       ;REPORT THE ERROR.
6145
6146 031770  005067  150312     60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6147
6148 031774                             ENDTST
     031774                                                                 L10044:
     031774  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
                                                                                T16::
                                         SETPRI #PRIO5          ;ALLOW THE LTC TO INTERRUPT.
                                                                                MOV     #PRIO5,R0
                                                                                TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
                                         MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (16)
                                         MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE WITHIN A TEST.
                                         MOV    #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
                                         MOV    #1601.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
                                         MOV    #EM1604,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
                                         CLR    ERSMRF          ;CLEAR THE ERROR SUMMARY FLAGS.
                                         MOV    #ERCNTB,R0
                                        JSR     PC,CLR16W       ;CLEAR THE ERROR COUNTER TABLE.
                                         CLR    EXOERR          ;CLEAR THE "EXIT ON ERROR" FLAG
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1601 <<<<<.
                                ;-
                                        JSR     PC,RESETT       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
                                        BCS     .+6             ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
                                        JMP     60$             ;YES,  EXIT THE TEST.
                                ;+
                                ; VERIFY READ/WRITE CAPABILITY TO INDIREC  ADDRESS FIELD OF CSR
                                ;-
                                         INC    ERRNBR          ;SET THE ERROR REPORT NUMBER TO 1602.
                                         MOV    #17,R2          ;SET LOOP COUNT.
                                         MOV    CSRA,R4         ;GET CSR ADDRESS.
                                2$:      MOV    R2,(R4)         ;WRITE COUNT TO CSR.
                                         MOV    (R4),R1         ;READ BACK THE CONTENTS OF THE CSR
                                         BIC    #177760,R1      ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
                                         CMP    R1,R2           ;CHECK FOR CORRECT DATA WRITTEN/READ.
                                        BEQ     4$              ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                    ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
                                         MOV    #ER1601,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
                                         CLR    R3              ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
                                         CLR    R5              ;CAUSE REPORT OF LINE 0.
                                        ERROR                   ;  >>>>> ERROR # 1602 <<<<<
                                                                                TRAP    C$ERROR
                                ;+
                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
                                ;-
                                         BIT    #BIT06,OPTION   ;HAS EXTENDED ERROR BEEN REQUESTED ?
                                        BEQ     60$             ;EXIT THE TEST IF IT HASN'T.

                                4$:      DEC    R2              ;DECREMENT LOOP COUNT (IND.ADD.REG ADDRESS).
                                        BGE     2$              ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
```

```
                                        ;+
                                        ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
                                        ;   ACTIVE LINES.  BEFORE WRITING EACH PATTERN, CLEAR ALL THE BITS.
                                        ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1603 - 1605 <<<<<.
                                        ;-
032154  005267  151610                          INC     ERRNBR          ;SET THE ERROR NUMBER TO 1603.
032160  005003                                  CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
032162  012704  000002                          MOV     #2,R4           ;INDICATE R/W ACCESS, CLEAR FIRST.
032166  004767  166704                        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.
                                        ;-
032172  005767  150106                          TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
032176  001017                                BNE     60$             ;EXIT IF IT IS.
                                        ;+
                                        ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
                                        ;   ACTIVE LINES.  BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
                                        ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1606 - 1608 <<<<<.
                                        ;-
032200  012767  003106  151562                  MOV     #1606.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
032206  005003                                  CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
032210  005404                                  NEG     R4              ;INDICATE R/W ACCESS, SET FIRST.
032212  004767  166660                        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.
                                        ;-
032216  005767  150062                          TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
032222  001005                                BNE     60$             ;EXIT IF IT IS.
                                        ;+
                                        ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
                                        ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1609 <<<<<
                                        ;-
032224  012767  003111  151536                  MOV     #1609.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
032232  004767  167072                        JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
032236  005067  150044            60$:        CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
032242                                        ENDTST
032242                                                                                  L10045:
032242  104401                                                                                  TRAP    C$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
                                                                                        T17::
                                         SETPRI #PRI05          ;ALLOW THE LTC TO INTERRUPT.
                                                                                        MOV     #PRI05,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.  EXOERF 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 <<<<<.
```

```
 6296                                           ;-
 6297 032370  012767  003252  151372            MOV     #1706.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
 6298 032376  005003                            CLR     R3              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
 6299 032400  005404                            NEG     R4              ;INDICATE R/M/W ACCESS, SET FIRST.
 6300 032402  004767  166470                    JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6301                                           ;+
 6302                                           ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
 6303                                           ; NOT BEEN REQUESTED; I.E.  EXOERR IS NON-ZERO.
 6304                                           ;-
 6305 032406  005767  147672                    TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
 6306 032412  001005                            BNE     60$             ;EXIT IF IT IS.
 6307
 6308                                           ;+
 6309                                           ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
 6310                                           ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1709 <<<<<
 6311                                           ;-
 6312 032414  012767  003255  151346            MOV     #1709.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
 6313 032422  004767  166702                    JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
 6314 032426  005067  147654            60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 6315 032432                                    ENDTST
      032432                                                                    L10046:
      032432  104401                                                                TRAP    C$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
                                                                                        T18::
                                         SETPRI #PRIO5          ;ALLOW THE LTC TO INTERRUPT.
032434                                                                                  MOV     #PRIO5,R0
032440  104441                                                                         TRAP    C$SPRI
        000022
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
032442  012767  000022  147654          MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (18)
032450  012767  177777  147630          MOV     #-1.,CTRLCF    ;INDICATE THAT WE ARE WITHIN A TEST.
032456  012767  000001  151302          MOV     #1,ERRTYP      ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
032464  012767  003411  151276          MOV     #1801.,ERRNBR  ;SET UP ERROR NUMBER IN THE ERROR TABLE.
032472  012767  012443  151272          MOV     #EM1801,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
032500  005067  147756                  CLR     ERSMRF         ;CLEAR THE ERROR SUMMARY FLAGS.
032504  012700  002464                  MOV     #ERCNTB,R0
032510  004767  165310                 JSR     PC,CLR16W      ;CLEAR THE ERROR COUNTER TABLE.
032514  005067  147564                  CLR     EXOERR         ;CLEAR THE "EXIT ON ERROR" FLAG.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1801 <<<<<.
                                ;-
032520  004767  166632                  JSR     PC,RESETT      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
032524  103402                          BCS     .+6            ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
032526  000167  000212                  JMP     60$            ;YES,  EXIT THE TEST.
032532  012767  003412  151230          MOV     #1802.,ERRNBR  ;SET THE ERROR REPORT NUMBER TO 1802.
                                ;+
                                ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR.
                                ; US  BYTE ACCESSES.
                                ;-
032540  012702  000017                  MOV     #17,R2         ;SET LOOP COUNT.
032544  016704  147472                  MOV     CSRA,R4        ;GET CSR ADDRESS.
032550  110214                 2$:      MOVB    R2,(R4)        ;WRITE COUNT TO CSR.
032552  111401                          MOVB    (R4),R1        ;READ BACK THE CONTENTS OF THE CSR
032554  042701  177760                  BIC     #177760,R1     ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
032560  020102                          CMP     R1,R2          ;CHECK FOR CORRECT DATA WRITTEN/READ.
032562  001412                          BEQ     4$             ;IS EXPECTED DATA BAD?  NO, SKIP ERROR REPORT.
                                    ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE  0 (D)."
032564  012767  016416  151202          MOV     #ER1601,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
032572  005003                          CLR     R3             ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
032574  005005                          CLR     R5             ;CAUSE REPORT OF LINE 0.
032576                                  ERROR                  ;  >>>>> ERROR # 1802 <<<<<
032576  104460                                                                         TRAP    C$ERROR
                                ;+
                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                ;-
032600  032767  000100  147416          BIT     #BIT06,OPTION  ;EXIT WITH TEST FAILURE MESSAGE IF
032606  001456                          BEQ     60$            ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                               ;DURING THE SOFTWARE QUESTIONS.
```

```
6370 032610  005302                  4$:     DEC     R2              ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
6371 032612  002356                          BGE     2$              ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
6372                                      ;+
6373                                      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6374                                      ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
6375                                      ;   EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6376                                      ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1803 - 1805 <<<<<.
6377                                      ;-
6378 032614  005267  151150                      INC     ERRNBR          ;SET THE ERROR NUMBER TO 1803.
6379 032620  012703  177777                      MOV     #-1,R3          ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6380 032624  012704  000002                      MOV     #2,R4           ;INDICATE R/W ACCESS, CLEAR FIRST.
6381 032630  004767  166242                      JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6382                                      ;+
6383                                      ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6384                                      ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
6385                                      ;-
6386 032634  005767  147444                      TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
6387 032640  001041                          BNE     60$             ;EXIT IF IT IS.
6388
6389                                      ;+
6390                                      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6391                                      ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
6392                                      ;   EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6393                                      ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1806 - 1808 <<<<<.
6394                                      ;-
6395 032642  012767  003416  151120            MOV     #1806.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6396 032650  005403                          NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6397 032652  004767  166220                      JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6398                                      ;+
6399                                      ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6400                                      ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
6401                                      ;-
6402 032656  005767  147422                      TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
6403 032662  001030                          BNE     60$             ;EXIT IF IT IS.
6404                                      ;+
6405                                      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6406                                      ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
6407                                      ;   EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6408                                      ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1809 - 1811 <<<<<.
6409                                      ;-
6410 032664  012767  003421  151076            MOV     #1809.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6411 032672  005403                          NEG     R3              ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6412 032674  005404                          NEG     R4              ;INDICATE R/W ACCESS, SET FIRST
6413 032676  004767  166174                      JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
6414                                      ;+
6415                                      ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6416                                      ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
6417                                      ;-
6418 032702  005767  147376                      TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
6419 032706  001016                          BNE     60$             ;EXIT IF IT IS.
6420                                      ;+
6421                                      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6422                                      ;   REGISTERS ON ALL ACTIVE LINES.  USE READ/WRITE ACCESSES.  BEFORE WRITING
6423                                      ;   EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6424                                      ;    REGTST ROUTINE REPORTS ERRORS WITH NUMBERS  >>>>> ERROR 1812 - 1814 <<<<<.
6425                                      ;-
6426 032710  012767  003424  151052            MOV     #1812.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
```

```
 6427 032716  005403                        NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
 6428 032720  004767  166152                JSR     PC,REGTST       ;WRITE AND VERIFY DATA PATTERNS.
 6429                                ;+
 6430                                ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
 6431                                ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
 6432                                ;-
 6433 032724  005767  147354                TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
 6434 032730  001005                        BNE     60$             ;EXIT IF IT IS.
 6435                                ;+
 6436                                ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
 6437                                ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1815 <<<<<
 6438                                ;-
 6439 032732  012767  003427  151030        MOV     #1815.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
 6440 032740  004767  166364                JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
 6441 032744  005067  147336          60$:  CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 6442 032750                                ENDTST
      032750                                                                L10047:
      032750  10440!                                                              TRAP    C$ETST
```

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

                                        BGNTST
                                                                                T19::
032752                           SETPRI #PRIO5          ;ALLOW THE LTC TO INTERRUPT.
032752  012700  000240                                                  MOV     #PRIO5,R0
032756  104441                                                         TRAP    C$SPRI
000023                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
032760  012767  000023  147336    MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (19)
032766  012767  177777  147312    MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
032774  012767  000001  150764    MOV     #1,ERRTYP      ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
033002  012767  003555  150760    MOV     #1901.,ERRNBR  ;SET UP ERROR NUMBER IN THE ERROR TABLE.
033010  012767  012517  150754    MOV     #EM1901,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
033016  005067  147440            CLR     ERSMRF         ;CLEAR THE ERROR SUMMARY FLAGS.
033022  012700  002464            MOV     #ERCNTB,R0
033026  004767  164772           JSR     PC,CLR16W      ;CLEAR THE ERROR COUNTER TABLE.
033032  005067  147246            CLR     EXOERR         ;CLEAR THE "EXIT ON ERROR" FLAG.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERRORS >>>>> 1901 <<<<<.
                                ;-
033036  004767  166314            JSR     PC,RESETT      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
033042  103402                   BCS     .+6            ;FATAL RESET ERROR?  NO, CONTINUE WITH TEST.
033044  000167  000136            JMP     60$            ;YES,  EXIT THE TEST.
033050  012767  003557  150712     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 <<<<<.
                                ;-
033056  005267  150706             INC     ERRNBR         ;SET THE ERROR NUMBER TO 1903.
033062  012703  177777             MOV     # 1,R3         ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
033066  012704  000001             MOV     #1,R4          ;INDICATE R/M/W ACCESS, CLEAR FIRST.
033072  004767  166000            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.
                                ;-
033076  005767  147202             TST     EXOERR         ;IS THE "EXIT ON ERROR" FLAG SET ?
033102  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 <<<<<.
                                        ;-
033104  012767  003562  150656                MOV     #1906.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
033112  005403                              NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
033114  004767  165756                      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.
                                        ;-
033120  005767  147160                      TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
033124  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 <<<<<.
                                        ;-
033126  012767  003565  150634                MOV     #1909.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
033134  005403                              NEG     R3              ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
033136  005404                              NEG     R4              ;INDICATE R/M/W ACCESS, SET FIRST.
033140  004767  165732                      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.
                                        ;-
033144  005767  147134                      TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
033150  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 <<<<<.
                                        ;-
033152  012767  003570  150610                MOV     #1912.,ERRNBR   ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
033160  005403                              NEG     R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
033162  004767  165710                      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.
                                        ;-
033166  005767  147112                      TST     EXOERR          ;IS THE "EXIT ON ERROR" FLAG SET ?
033172  001005                            BNE     60$             ;EXIT IF IT IS.
                                        ;+
                                        ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
                                        ;   THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1915 <<<<<
                                        ;-
033174  012767  003573  150566                MOV     #1915.,ERRNBR   ;SET UP ERROR NUMBER FOR NEXT RTN.
033202  004767  166122                      JSR     PC,REPSMR       ;REPORT ERROR SUMMARY IF NECESSARY.
033206  005067  147074            60$:      CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
033212                                      ENDTST
033212                                                                              L10050:
033212  104401                                                                              TRAP    C$ETST
```

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

 033214                                         BGNTST
 033214                                                                                 T20::
 033214                                          SETPRI #PRIO5          ;ALLOW THE LTC TO INTERRUPT.
 033214  012700  000240                                                                         MOV     #PRIO5,R0
 033220  104441                                                                                 TRAP    C$SPRI
         000024                                  INUM == INUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 033222  012767  000024  147074                 MOV     #INUM,TSTNUM    ;SET UP THE TEST NUMBER.       (20)
 033230  012767  177777  147050                 MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 033236  012767  000001  150522                 MOV     #1,ERRTYP       ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
 033244  012767  003721  150516                 MOV     #2001.,ERRNBR   ;SET UP ERROR NUMBER IN THE ERROR TABLE.
 033252  012767  012602  150512                 MOV     #EM2001,ERRMSG  ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
                                        ;+
                                        ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                        ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                        ; THIS SUBROUTINE REPORTS ERRORS >>>>> 2001 <<<<<.
                                        ;-
 033260  004767  166072                         JSR     PC,RESETT       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
 033264  103016                                 BCC     60$             ;FATAL RESET ERROR?  YES,  EXIT THE TEST.
                                        ;+
                                        ; READ THE STAT REGISTER ID BIT AND VERIFY THAT IT IS CLEAR.
                                        ;-
 033266  017701  146756                         MOV     @FSLSA,R1       ;READ THE STAT REGISTER CONTENTS.
 033272  032701  000400                         BIT     #BIT8,R1        ;CHECK THE ID BIT.
 033276  001011                                 BNE     60$             ;ID BIT SET?  YES, EXIT THE TEST.
 033300  012767  003722  150462                 MOV     #2002.,ERRNBR   ;NO,  SET THE ERROR REPORT NUMBER TO 2002.
 033306  012701  012652                         MOV     #EM2002,R1      ;GET THE PROPER ERROR MESSAGE.
 033312  012767  016136  150454                 MOV     #ER0503,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
 033320                                         ERROR                   ;ERROR NUMBER >>>>> 2002 <<<<<
 033320  104460                                                                                 TRAP    C$ERROR
 033322  005067  146760                 60$:    CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
 033326                                         ENDTST
 033326                                                                                 L10051:
 033326  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.
                                        ;*
                                        ;-- ***********************************************************************

 033330                                         BGNTST
 033330                                                                                  T21::
 033330                                         SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
 033330  012700  000240                                                                          MOV     #PRIO5,R0
 033334  104441                                                                                  TRAP    C$SPRI
         000025                                  TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 033336  012767  000025  146760                  MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.       (23)
 033344  012767  177777  146734                  MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
 033352  012767  000001  150406                  MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 033360  012767  004375  150402                  MOV     #2301.,ERRNBR  ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
 033366  012767  012725  150376                  MOV     #EM2301,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
 033374  012767  017222  150372                  MOV     #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                        ;+
                                        ; RESET THE DUT TO A KNOWN STATE. REMOVE THE STATUS CODES FROM THE FIFO.
                                        ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                        ; THIS SUBROUTINE REPORTS ERROR >>>>> 2301 <<<<<.
                                        ;-
 033402  004767  164374                          JSR     PC,CLNRST      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
 033406  103114                                  BCC     60$            ;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.
                                        ;-
 033410  016705  146614                          MOV     ACTLNS,R5      ;PASS THE ACTIVE LINE BIT MAP.
 033414  012700  000200                          MOV     #200,R0        ;PASS THE LNCTRL CONTENTS.
 033420  004767  170126                          JSR     PC,WTWLNC      ;INITIALISE THE LNCTRL REGISTERS.
 033424  012700  177670                          MOV     #177670,R0     ;PASS THE LPR CONTENTS.
 033430  004767  170146                          JSR     PC,WTWLPR      ;INITIALSE THE LPR REGISTERS ON ALL LINES.
 033434  012704  000012                          MOV     #10.,R4        ;PASS DELAY TIME OF 10 MILLI-SECONDS.
 033440  004767  164460                          JSR     PC,DELAY       ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
 033444  012705  177777                          MOV     #MAPLNS,R5     ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
 033450  004767  167224                          JSR     PC,TXENBL      ;ENABLE TRANSMITTERS ON ALL LINES.
                                        ;+
                                        ; TEST ALL ACTIVE LINES INDIVIDUALLY.
                                        ; DISABLE TRANSMISSION ON EACH ACTIVE LINE.
                                        ;
 033454  012703  000001                          MOV     #1,R3          ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
 033460  005004                                  CLR     R4             ;CLEAR THE LINE NUMBER COUNTER.
 033462  012767  004376  150300  2$:             MOV     #2302.,ERRNBR  ;SET THE ERROR NUMBER TO 2302.
 033470  030367  146534                          BIT     R3,ACTLNS      ;CHECK IF THE LINE IS ACTIVE.
 033474  001453                                  BEQ     6$             ;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.
                                                ;-
```

HARDWARE TEST TXFNB1

```
6641 033476  010305                    MOV     R3,R5           ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6642 033500  004767  167100             JSR     PC,TXDSBL       ;DISABLE TRANSMISSION ON THE LINE UNDER TEST.
6643 033504  010477  146536             MOV     R4,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
6644 033510  005777  146542             TST     @TXAD2A         ;VERIFY THE TX ENABLE BIT IS SET.
6645 033514  100433                    BMI     4$              ;GO REPORT ERROR IF TX ENABLE BIT SET.
6646                                   ;+
6647                                   ; WRITE DATA BYTE (ASCII <LF>) TO THE OUTPUT FIFO.
6648                                   ; WAIT FOR A TX ACTION TO BE RETURNED, REPORT ERROR IF A TX ACTION
6649                                   ; IS FOUND BEFORE TIME OUT OCCURS.
6650                                   ;-
6651 033516  012767  004577  150244     MOV     #2303.,ERRNBR   ;SET ERROR NUMBER TO 2303.
6652 033524  112777  000012  146516     MOVB    #12,@FDATA      ;WRITE THE DATA BYTE TO THE DUT'S OUTPUT FIFO.
6653 033532  012701  170003             MOV     #170003,R1      ;TEST BIT 15, TIMEOUT OF 3 MILLI SECS.
6654 033536  016702  146500             MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6655 033542  004767  167522             JSR     PC,WAIBIS       ;WAIT FOR TX ACTION TO COME BACK.
6656 033546  103416                    BCS     4$              ;GO REPORT ERROR IF A TX-ACTION FOUND.
6657                                   ;+
6658                                   ; WAIT FOR THE DATA TO APPEAR IN THE FIFO, REPORT ERROR IF DATA FOUND.
6659                                   ;-
6660 033550  005267  150214             INC     ERRNBR          ;SET ERROR NUMBER TO 2304.
6661 033554  012701  070012             MOV     #70012,R1       ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
6662 033560  016702  146456             MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6663 033564  004767  167500             JSR     PC,WAIBTS       ;WAIT FOR RX DATA AVAILABLE TO SET.
6664 033570  103405                    BCS     4$              ;REPORT ERROR IF DATA RECEIVED IN THE FIFO.
6665 033572  005267  150172             INC     ERRNBR          ;SET ERROR NUMBER TO 2305.
6666 033576  017702  146442             MOV     @RBUFA,R2       ;READ THE DATA FROM THE FIFO.
6667 033602  100010                    BPL     6$              ;SKIP ERROR REPORT IF DATA ISN'T THERE.
6668
6669 033604  010401           4$:      MOV     R4,R1           ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6670 033606  012702  012772             MOV     #EM2302,R2      ;PASS THE MESSAGE TO BE REPORTED.
6671                                                           ;"TX ENABLE BIT BAD ON LINE: NN".
6672 033612                             ERROR                   ;                >>>>> ERROR <<<<<.
     033612  104460                                                                   TRAP    C$ERROR
6673
6674 033614  032767  000100  146402     BIT     #BIT06,OPTION   ;EXIT THE TEST, WITH THE TEST FAILED MESSAGE,
6675 033622  001406                    BEQ     60$             ;IF EXTENDED ERROR REPORTING HAS NOT BEEN
6676                                                           ;REQUESTED.
6677
6678                                   ;+
6679                                   ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6680                                   ;
6681 033624  000241           6$:      CLC                     ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
6682 033626  006103                    ROL     R3              ;SHIFT THE BIT MAP FOR THE NEXT LINE.
6683 033630  005204                    INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
6684 033632  020427  000020             CMP     R4,#NUMLNS      ;HAVE ALL THE LINES BEEN TESTED?.
6685 033636  002711                    BLT     2$              ;NO; BRANCH TO TEST THE NEXT LINE.
6686
6687 033640  005067  146442   60$:      CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6688 033644                             ENDTST
     033644                                                                   L10052:
     033644  104401                                                                 TRAP    C$ENDT
```

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

033646                                  BGNTST
033646                                                                  T22::
033646                          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
033646  012700  000240                                                  MOV     #PRI05,R0
033652  104441                                                          TRAP    C$SPRI
000026                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
033654  012767  000026  146442    MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (24)
033662  012767  177777  146416    MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
033670  012767  000001  150070    MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
033676  012767  004541  150064    MOV     #2401.,ERRNBR  ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
033704  012767  013030  150060    MOV     #EM2401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
033712  012767  017222  150054    MOV     #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 2401 <<<<<.
                                ;-
033720  004767  164056                  JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
033724  103133                        BCC     60$             ;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.
                                ;-
033726  016705  146276                  MOV     ACTLNS,R5       ;PASS THE ACTIVE LINE BIT MAP.
033732  012700  000200                  MOV     #200,R0         ;PASS THE LNCTRL CONTENTS.
033736  004767  167610                  JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTERS.
033742  012700  177670                  MOV     #177670,R0      ;PASS THE LPR CONTENTS.
033746  004767  167670                  JSR     PC,WTWLPR       ;INITIALSE THE LPR REGISTERS ON ALL LINES.
033752  012704  000012                  MOV     #10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
033756  004767  164142                  JSR     PC,DELAY        ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
033762  012705  177777                  MOV     #MAPLNS,R5      ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
033766  004767  166612                  JSR     PC,TXDSBL       ;DISABLE TRANSMITTERS ON ALL LINES.
                                ;*
                                ; TEST ALL ACTIVE LINES INDIVIDUALLY.
                                ; ENABLE TRANSMISSION ON EACH ACTIVE LINE.
                                ;-
033772  012703  000001                  MOV     #1,R3           ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
033776  005004                          CLR     R4              ;CLEAR THE LINE NUMBER COUNTER.
034000  012767  004542  147762  2$:     MOV     #2402.,ERRNBR   ;SET THE ERROR NUMBER TO 2402.
034006  030367  146216                  BIT     R3,ACTLNS       ;CHECK IF THE LINE IS ACTIVE.
034012  001467                          BEQ     8$              ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
                                ;*
                                ; SELECT THE LINE UNDER TEST.
                                ;  SET THE TX_ENABLE BIT IN TBUFADR REGISTER.
                                ;  VERIFY IT IS SET, REPORT ERROR IF CLEAR.
                                ;-
```

```
6744 034014  010305                        MOV     R3,R5           ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6745 034016  004767  166656                JSR     PC,TXENBL       ;ENABLE TRANSMISSION ON THE LINE UNDER TEST.
6746 034022  012705  000012                MOV     #10.,R5         ;SET TXCHAR/LOOP COUNT TO 10.
6747 034026  010477  146210                MOV     R4,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
6748 034032  005777  146220                TST     @TXAD2A         ;VERIFY THE TX_ENABLE BIT IS SET.
6749 034036  100045                        BPL     6$              ;GO REPORT ERROR IF TX_ENABLE BIT CLEAR.
6750                                       ;+
6751                                       ; WRITE DATA BYTE (ASCII <LF>) TO OUTPUT FIFO.
6752                                       ; WAIT FOR A TX_ACTION TO BE RETURNED, REPORT ERROR IF NO TX_ACTION
6753                                       ; FOUND BEFORE TIME-OUT OCCURS.
6754                                       ;-
6755 034040  012767  004543  147722  4$:   MOV     #2403.,ERRNBR   ;SET ERROR NUMBER TO 2403.
6756 034046  112777  000012  146174        MOVB    #12,@FDATA      ;WRITE THE DATA BYTE TO THE DUT'S OUTPUT FIFO.
6757 034054  012701  170004                MOV     #170004,R1      ;TEST BIT 15, TIMEOUT OF 4 MILLI SECS.
6758 034060  016702  146156                MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6759 034064  004767  167200                JSR     PC,WAIBIS       ;WAIT FOR TX_ACTION TO COME BACK.
6760 034070  103030                        BCC     6$              ;GO REPORT ERROR IF NO TX-ACTION FOUND.
6761                                       ;+
6762                                       ; WAIT FOR THE DATA TO APPEAR IN THE FIFO, REPORT ERROR IF TIME-OUT.
6763                                       ;-
6764 034072  005267  147672                INC     ERRNBR          ;SET ERROR NUMBER TO 2404.
6765 034076  012701  070012                MOV     #70012,R1       ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
6766 034102  016702  146134                MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6767 034106  004767  167156                JSR     PC,WAIBIS       ;WAIT FOR RX_DATA_AVAILABLE TO SET.
6768 034112  103017                        BCC     6$              ;REPORT ERROR IF NO DATA RECEIVED IN THE FIFO.
6769 034114  005267  147650                INC     ERRNBR          ;SET ERROR NUMBER TO 2405.
6770 034120  017702  146120                MOV     @RBUFA,R2       ;READ THE DATA FROM THE FIFO.
6771 034124  100012                        BPL     6$              ;GO REPORT ERROR IF THER IS'NT ANY DATA THERE.
6772 034126  005267  147636                INC     ERRNBR          ;SET ERROR NUMBER TO 2406.
6773 034132  000302                        SWAB    R2              ;PUT THE LINE NUMBER IN THE LOW BYTE.
6774 034134  042702  177760                BIC     #177760,R2      ;CLEAR THE UNWANTED BITS.
6775 034140  020204                        CMP     R2,R4           ;DID THE DATA COME FROM THE CORRECT LINE?.
6776 034142  001003                        BNE     6$              ;NO; GO REPORT THE ERROR.
6777 034144  005305                        DEC     R5              ;DECREMENT THE TXCHAR/LOOP COUNTER.
6778 034146  001334                        BNE     4$              ;LOOP TO TX THE NEXT CHAR.
6779 034150  000410                        BR      8$              ;GO TEST THE NEXT LINE.
6780
6781 034152  010401                 6$:    MOV     R4,R1           ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6782 034154  012702  012772                MOV     #EM2302,R2      ;PASS THE MESSAGE TO BE REPORTED.
6783                                                               ;"TX_ENABLE BIT BAD ON LINE: NN".
6784 034160                                ERROR                   ;                  >>>>> ERROR <<<<<.
     034160  104460                                                                      TRAP    C$ERROR
6785
6786 034162  032767  000100  146034        BIT     #BIT06,OPTION   ;EXIT THE TEST IF EXTENDED ERROR REPORTING
6787 034170  001411                        BEQ     60$             ;HAS NOT BEEN ENABLED, SINCE THE TEST HAS FAILED.
6788
6789                                 ;+
6790                                 ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6791                                 ;
6792 034172  010305                 8$:    MOV     R3,R5           ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6793 034174  004767  166404                JSR     PC,TXDSBL       ;CLEAR THE TX_ENABLE BIT ON THIS LINE.
6794 034200  000241                        CLC                     ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
6795 034202  006103                        ROL     R3              ;SHIFT THE BIT MAP FOR THE NEXT LINE.
6796 034204  005204                        INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
6797 034206  020427  000020                CMP     R4,#NUMLNS      ;HAVE ALL THE LINES BEEN TESTED?.
6798 034212  002672                        BLT     2$              ;NO; BRANCH TO TEST THE NEXT LINE.
6799
```

```
6800 034214  005067  146066          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6801 034220                                  ENDTST
     034220                                                          L10053:
     034220  104401                                                          TRAP    C$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.
                                        ;*
                                        ;-- ******************************************************************

 034222                                         BGNTST
 034222                                                                                 T23::
 034222                                         SETPRI  #PRI05          ;ALLOW THE LTC TO INTERRUPT.
 034222  012700  000240                                                                         MOV     #PRI05,R0
 034226  104441                                                                                 TRAP    C$SPRI
         000027                                   TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 034230  012767  000027  146066                   MOV     #TNUM,TSTNUM  ;SET UP THE TEST NUMBER.          (26)
 034236  012767  177777  146042                   MOV     #-1,CTRLCF    ;INDICATE THAT WE ARE IN A TEST.
 034244  012767  000001  147514                   MOV     #1,ERRTYP     ;SET ERROR FATAL ERROR TYPE IN ERROR TABLE.
 034252  012767  003101  147510                   MOV     #1601.,ERRNBR ;SET FIRST ERROR REPORT NUMBER IN ERROR TABLE.
 034260  012767  013073  147504                   MOV     #EM2601.,ERRMSG ;SET TEST ERROR MESSAGE IN ERROR TABLE.
                                        ;+
                                        ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                        ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                        ; THIS SUBROUTINE REPORTS ERRORS FROM >>>>> 2601 THRU 2602 <<<<<.
                                        ;-
 034266  004767  165064                         JSR     PC,RESETT       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
 034272  103402                                 BCS     2$              ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
 034274  000167  001044                         JMP     60$             ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
 034300  012767  005053  147462 2$:             MOV     #2603.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 2603.
                                        ;+
                                        ; ENABLE TRANSMITTERS ON ALL LINES.
                                        ;-
 034306  012705  177777          4$:            MOV     #MAPLNS,R5      ;PASS ACTIVE LINE BIT MAP.
 034312  004767  166362                         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.
                                        ;-
 034316  005067  145772                         CLR     RXINIC          ;CLEAR THE RX INTERRUPT COUNTER.
 034322  005067  145770                         CLR     RXINIF          ;CLEAR THE RX INTERRUPT FLAGS.
 034326  005067  145774                         CLR     TXINIC          ;CLEAR THE TX INTERRUPT COUNTER.
 034332  012767  002726  145742                 MOV     #BUFBAS,BUFPTR  ;LOAD THE BUFFER PTR WITH THE BUFFER BASE ADR.
 034340                                         SETVEC  RXVECA,#RXINPT,#PRI06 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
 034340  012746  000300                                                                         MOV     #PRI06,-(SP)
 034344  012746  024036                                                                         MOV     #RXINPT,-(SP)
 034350  016746  145656                                                                         MOV     RXVECA,-(SP)
 034354  012746  000003                                                                         MOV     #3,-(SP)
 034360  104437                                                                                 TRAP    C$SVEC
 034362  062706  000010                                                                         ADD     #10,SP
 034366                                         SETVEC  TXVECA,#CACHTX,#PRI06 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
 034366  012746  000300                                                                         MOV     #PRI06,-(SP)
 034372  012746  023660                                                                         MOV     #CACHTX,-(SP)
```

```
        034376  016746  145632                                                          MOV     TXVECA,-(SP)
        034402  012746  000003                                                          MOV     #3,-(SP)
        034406  104437                                                                  TRAP    C$SVEC
        034410  062706  000010                                                          ADD     #10,SP
  6849  034414                      SETPRI  #PRI04          ;ALLOW DEVICE INTERRUPTS.
        034414  012700  000200                                                          MOV     #PRI04,R0
        034420  104441                                                                  TRAP    C$SPRI
  6850                          ;+
  6851                          ;ENABLE RECEPTION INTERRUPTS.
  6852                          ;DELAY 4 MS TO ALLOW TIME FOR THE INTERRUPTS TO TAKE PLACE.
  6853                          ;DISABLE RECEPTION INTERRUPTS.
  6854                          ;-
  6855  034422  004767  165624          JSR     PC,RXIE1        ;ENABLE THE RECEPTION INTERRUPTS.
  6856  034426  012704  000004           MOV     #4,R4           ;PASS 4 MS COUNT TO THE DELAY ROUTINE.
  6857  034432  004767  163466          JSR     PC,DELAY        ;DELAY 4 MILLI-SECONDS.
  6858  034436  004767  165550          JSR     PC,RXIE0        ;DISABLE RECEPTION INTERRUPTS.
  6859                          ;+
  6860                          ; VERIFY THAT THE CORRECT INTERRUPTS TOOK PLACE.
  6861                          ;    TEST THE INT COUNTER TO VERIFY THAT INTERRUPTS TOOK PLACE.
  6862                          ;-
  6863  034442  005767  145646           TST     RXINTC          ;CHECK THE RX INTERRUPT COUNT.
  6864  034446  001017                   BNE     6$              ;SKIP THE FOLLOWING ERRORS IF COUNT <> 0.
  6865                          ;+
  6866                          ; DETERMINE REASON FOR NO RX INTERRUPTS AND PRINT PROPER ERROR MESSAGE.
  6867                          ;-
  6868  034450  012701  013310           MOV     #EM2604,R1      ;SET UP MSG IN CASE "RX.DATA.AVAIL IS CLR".
  6869  034454  032777  000200  145560   BIT     #BIT7,@CSRA     ;TEST THE RX.DATA.AVAIL BIT OF THE CSR.
  6870  034462  001416                   BEQ     8$              ;GO REPORT ERROR IF RX.DATA.AVAIL IS CLR.
  6871  034464  012701  013222           MOV     #EM2603,R1      ;SET UP MSG IN CASE "DATA.VALID IS CLEAR".
  6872  034470  032777  100000  145546   BIT     #BIT15,@RBUFA   ;TEST THE DATA.VALID BIT OF THE FIFO.
  6873  034476  001410                   BEQ     8$              ;GO REPORT ERROR IF DATA.VALID IS CLEAR.
  6874  034500  012701  013131           MOV     #EM2602,R1      ;SET UP MSG,"DATA.VALID IS SET".
  6875  034504  000405                   BR      8$              ;GO REPORT THE ERROR.
  6876                          ;+
  6877                          ; IF RX INTS OCCURRED WITH RX.DATA.AVAIL CLEAR, REPORT THE ERROR.
  6878                          ;-
  6879  034506  005767  145604    6$:    TST     RXINTF          ;CHECK THE RX INTERRUPT FLAGS.
  6880  034512  100014                   BPL     10$             ;SKIP THE ERROR IF FLAG IS CLEAR.
  6881  034514  012701  013404           MOV     #EM2605,R1      ;SET UP THE PROPER MESSAGE.
  6882                          ;+
  6883                          ; REPORT THE ERROR WHICH HAS BEEN FOUND.
  6884                          ;-
  6885  034520                    8$:    ERRDF   2603,EM2601,ER0503;             >>>>> ERROR #2603 <<<<<
        034520  104455                                                                  TRAP    C$ERDF
        034522  005053                                                                  .WORD   2603
        034524  013073                                                                  .WORD   EM2601
        034526  016136                                                                  .WORD   ER0503
  6886
  6887                          ;+
  6888                          ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
  6889                          ;-
  6890
  6891  034530  032767  000100  145466           BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
  6892  034536  001002                   BNE     .+6             ;
  6893  034540  000167  000556           JMP     34$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
  6894
  6895                          ;+
```

```
                                        ; VERIFY THAT NO TX INTERRUPTS HAVE BEEN GENERATED SO FAR IN THIS TEST.
                                        ;-
034544  016702  145556                  10$:    MOV     TXINTC,R2       ;LOAD # OF TX INTERRUPTS FOR ER0504 RTN.
034550  001414                                  BEQ     12$             ;SKIP ERROR IF NO TX INTERRUPTS.
                                            ;REPORT "TX INTERRUPTS(S) RECEIVED WITH TX INTERRUPTS DISABLED."
034552  012701  007505                          MOV     #EM0526,R1      ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
034556                                          ERRDF   2604,EM2601,ER0504;             >>>>> ERROR #2604 <<<<<.
034556  104455                                                                          TRAP    C$ERDF
034560  005054                                                                          .WORD   2604
034562  013073                                                                          .WORD   EM2601
034564  016174                                                                          .WORD   ER0504
                                        ;+
                                        ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                        ;-
034566  032767  000100  145430                BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
034574  001002                                BNE     .+6             ;
034576  000167  000520                        JMP     34$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                        ;+
                                        ; CLEAN OUT THE INTERRUPT VECTORS USED IN THIS TEST.
                                        ;-
034602                                  12$:    SETPRI  #PRI06          ;DISABLE DEVICE INTERRUPTS.
034602  012700  000300                                                                  MOV     #PRI06,R0
034606  104441                                                                          TRAP    C$SPRI
034610                                          CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
034610  016700  145416                                                                  MOV     RXVECA,R0
034614  104436                                                                          TRAP    C$CVEC
034616                                          CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
034616  016700  145412                                                                  MOV     TXVECA,R0
034622  104436                                                                          TRAP    C$CVEC
                                        ;+
                                        ; TEST TRANSMISSION INTERRUPTS.
                                        ;   SET UP FOR RX AND TX INTERRUPTS:
                                        ;   RX INTERRUPT SERVICE ROUTINE COUNTS RX INTERRUPTS.
                                        ;   TX INTERRUPT SERVICE ROUTINE COUNTS THE INTERRUPT AND SETS FLAGS.
                                        ;-
034624  005067  145464                          CLR     RXINTC          ;CLEAR THE RX INTERRUPT COUNTER.
034630  005067  145472                          CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNTER.
034634  005067  145470                          CLR     TXINTF          ;CLEAR THE RX INTERRUPT FLAGS.
034640                                          SETVEC  RXVECA,#CACHRX,#PRI06 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
034640  012746  000300                                                                  MOV     #PRI06,-(SP)
034644  012746  023632                                                                  MOV     #CACHRX,-(SP)
034650  016746  145356                                                                  MOV     RXVECA,-(SP)
034654  012746  000003                                                                  MOV     #3,-(SP)
034660  104437                                                                          TRAP    C$SVEC
034662  062706  000010                                                                  ADD     #10,SP
034666                                          SETVEC  TXVECA,#TXINTR,#PRI06 ;SET UP INT VECTOR TO TX INT ROUTINE.
034666  012746  000300                                                                  MOV     #PRI06,-(SP)
034672  012746  024146                                                                  MOV     #TXINTR,-(SP)
034676  016746  145332                                                                  MOV     TXVECA,-(SP)
034702  012746  000003                                                                  MOV     #3,-(SP)
034706  104437                                                                          TRAP    C$SVEC
034710  062706  000010                                                                  ADD     #10,SP
034714                                          SETPRI  #PRI04          ;ALLOW DEVICE INTERRUPTS.
034714  012700  000200                                                                  MOV     #PRI04,R0
```

```
     034720  104441                                                TRAP     C$SPRI
                                 ;+
                                 ; VERIFY THAT THE TX_ACTION BIT IS CLEAR.
                                 ;-
     034722  012705  000022              MOV    #18.,R5          ;INITIALIZE THE LOOP COUNTER.
     034726  012701  000144              MOV    #100.,R1         ;SET 100 MS TIME-OUT.
     034732  012702  100000              MOV    #BIT15,R2        ;SELECT TX_ACTION BIT TO TEST.
     034736  016704  145300              MOV    CSRA,R4          ;PASS DUT CSR AS THE WORD TO TEST.
     034742  012703  100000      14$:    MOV    #BIT15,R3        ;WAIT FOR TX_ACTION TO BE SET.
     034746  004767  163326              JSR    PC,MSLOOP        ;WAIT UP TO 100 MS FOR TX_ACTION SET.
     034752  103026                      BCC    20$              ;IF TIME-OUT, CONSIDER TX_ACTION CLEAR.
     034754  005003                      CLR    R3               ;NOW, WAIT FOR TX_ACTION CLEAR.
     034756  004767  163316              JSR    PC,MSLOOP        ;WAIT UP TO 100 MS FOR TX_ACTION CLEAR.
     034762  103005                      BCC    16$              ;IF TIME-OUT, REPORT TX_ACTION WON'T CLEAR.
     034764  005305                      DEC    R5               ;DECREMENT THE TX_ACTION SET COUNTER.
     034766  001365                      BNE    14$              ;LOOP IF NOT TOO MANY TX_ACTIONS FOUND.
                                 ;REPORT "TX_ACTION SET REPEATEDLY AFTER RESET, NO DATA SENT."
     034770  012701  013526              MOV    #EM2607,R1       ;SELECT ERROR MESSAGE.
     034774  000402                      BR     18$              ;GO TO REPORT THE ERROR.
     034776  012701  013622      16$:    MOV    #EM2608,R1       ;SELECT TX_ACTION STUCK SET MSG.
     035002                      18$:    ERRDF  2605,EM2606,ER0503;         >>>>> ERROR #2605 <<<<<.
     035002  104455                                                TRAP     C$ERDF
     035004  005055                                                .WORD    2605
     035006  013467                                                .WORD    EM2606
     035010  016136                                                .WORD    ER0503

                                 ;+
                                 ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                 ;-
     035012  032767  000100  145204      BIT    #BIT06,OPTION    ;EXIT WITH TEST FAILURE MESSAGE IF
     035020  001540                      BEQ    34$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
     035022  004767  166006              JSR    PC,TXIE1         ;ENABLE TX INTERRUPTS FOR THE TX_INT TESTING.
     035026  000430                      BR     24$              ;GO TO TEST WITH TX_ACTION SET.
                                 ;+
                                 ; VERIFY THAT NO INTERRUPTS OCCUR WITH TX_ACTION CLEAR.
                                 ;-
     035030  004767  166000      20$:    JSR    PC,TXIE1         ;ENABLE TX_INTERRUPTS.
     035034  012704  000062              MOV    #50.,R4          ;PASS 50 MS TIME TO THE DELAY ROUTINE.
     035040  004767  163060              JSR    PC,DELAY         ;DELAY 50 MILLI-SECONDS TO ALLOW INTS TO OCCUR.
     035044  005767  145256              TST    TXINTC           ;TEST THE TX INTERRUPT COUNT.
     035050  001417                      BEQ    24$              ;SKIP THE ERROR IF NO TX INTERRUPTS.
     035052  012701  013526              MOV    #EM2607,R1       ;SELECT MESSAGE IN CASE TX INT FLAG CLEAR.
     035056  005767  145246              TST    TXINTF           ;TEST THE TX INTERRUPT FLAGS.
     035062  100002                      BPL    22$              ;GO REPORT ERROR IF TX FLAG IS CLEAR.
     035064  012701  013673              MOV    #EM2609,R1       ;TX FLAG IS SET, SELECT PROPER ERROR MESSAGE.
                                 ;REPORT "TRANSMIT INTERRUPT TEST ERROR:..."
     035070                      22$:    ERRDF  2606,EM2606,ER0503;         >>>>> ERROR #2606 <<<<<.
     035070  104455                                                TRAP     C$ERDF
     035072  005056                                                .WORD    2606
     035074  013467                                                .WORD    EM2606
     035076  016136                                                .WORD    ER0503

                                 ;+
                                 ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                 ;-
     035100  032767  000100  145116      BIT    #BIT06,OPTION    ;EXIT WITH TEST FAILURE MESSAGE IF
     035106  001500                      BEQ    32$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
```

```
                                        ;+
                                        ; PREPARE TX INTERRUPT COUNTER AND FLAGS.
                                        ;-
035110  005067  145212          24$:      CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNT.
035114  005067  145210                    CLR     TXINTF          ;CLEAR THE TX INTERRUPT FLAGS.
                                        ;+
                                        ; SET UP LINE PARAMETERS FOR TRANSMISSION.
                                        ;-
035120  012705  177777                    MOV     #MAPLNS,R5      ;PASS ACTIVE LINES BIT MAP.
035124  012700  000200                    MOV     #200,R0         ;PASS INERT STATE, INTERNAL LOOPBACK.
035130  004767  166416                    JSR     PC,WTWLNC       ;DISABLE RECPTION AND DMA, ETC. ON DUT.
035134  012700  156430                    MOV     #156430,R0      ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
035140  004767  166436                    JSR     PC,WTWLPR       ;WRITE TO ALL LPR REGISTERS.
                                        ;+
                                        ; SEND A NULL CHAR TO EACH LINE.
                                        ;-
035144  016701  145140                    MOV     IESTAT,R1       ;SET UP THE STATE OF THE INTERRUPT ENABLE BITS.
035150  005002                            CLR     R2              ;CLEAR THE LINE COUNTER.
035152  010177  145064          25$:      MOV     R1,@CSRA        ;SET UP THE LINE NUMBER AND INTERRUPT ENABLE
                                                                  ;BITS IN THE CSR.
035156  112777  000000  145064            MOVB    #0,@FDATA       ;SEND A NULL CHARACTER TO THE OUTPUT FIFO.
035164  005201                            INC     R1              ;NEXT CSR CONTENTS.
035166  005202                            INC     R2              ;NEXT LINE.
035170  020227  000020                    CMP     R2,#NUMLNS      ;IF ALL LINES HAVE NOT BEEN SERVICED THEN
035174  002766                            BLT     25$             ;BRANCH.
                                        ;+
                                        ; DELAY 250 MILLI-SECONDS TO ALLOW INTERRUPTS TO OCCUR.
                                        ;-
035176  012704  000372                    MOV     #250.,R4        ;SET UP FOR 250 MS DELAY.
035202  004767  162716                    JSR     PC,DELAY        ;WAIT 250 MS.
                                        ;+
                                        ; VERIFY THAT TX INTERRUPTS OCCURRED.
                                        ;-
035206  005767  145114                    TST     TXINTC          ;CHECK THE TX INTERRUPT COUNTER.
035212  001010                          BNE     26$               ;SKIP THE FOLLOWING ERROR IF WE GOT TX INTS.
                                        ;+
                                        ; DETERMINE THE REASON THAT WE RECEIVED NO INTERRUPTS.
                                        ;-
035214  012701  013752                    MOV     #EM2610,R1      ;SET UP MSG IN CASE "TX_ACTION IS SET".
035220  005777  145016                    TST     @CSRA           ;CHECK THE DUT CSR.
035224  100410                          BMI     28$               ;GO TO REPORT ERROR IF TX_ACTION IS SET.
035226  012701  014044                    MOV     #EM2611,R1      ;SET UP "TX_ACTION NOT SET" MESSAGE.
035232  000405                          BR      28$               ;GO AND REPORT THE ERROR.
                                        ;+
                                        ; CHECK TO VERIFY THAT TX_ACTION WAS SET FOR EACH INTERRUPT.
                                        ;-
035234  005767  145070          26$:      TST     TXINTF          ;CHECK THE TX INTERRUPT FLAGS.
035240  100012                          BPL     30$               ;SKIP ERROR IF TX_ACTION CLR FLAG IS CLEAR.
035242  012701  013673                    MOV     #EM2609,R1      ;SET UP TX INT WITH "TX_ACTION CLR" MSG.
                                        ;+
                                        ; REPORT "TRANSMIT INTERRUPT TEST ERROR;...."
                                        ;-
035246                          28$:      ERRDF   2607,EM2606,ER0503;          >>>>> ERROR #2607 <<<<<
035246  104455                                                                    TRAP    C$ERDF
035250  005057                                                                    .WORD   2607
035252  013467                                                                    .WORD   EM2606
```

M14


```
 035254  016136                                                                  .WORD   ER0503

                                        ;+
                                        ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                        ;-
 035256  032767  000100  144740            BIT     #BIT06,OPTION    ;EXIT WITH TEST FAILURE MESSAGE IF
 035264  001411                            BEQ     32$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED

                                   ;+
                                   ; VERIFY THAT NO RX INTERRUPTS HAVE BEEN GENERATED SO FAR IN THIS TEST.
                                   ;-
 035266  016702  145022            30$:    MOV     RXINTC,R2        ;LOAD # OF RX INTERRUPTS FOR ER0504 RTN.
 035272  001406                            BEQ     32$              ;SKIP ERROR IF NO RX INTERRUPTS.
 035274  012701  007415                    MOV     #EM0525,R1       ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
                                        ;REPORT "RX INTERRUPTS(S) RECEIVED WITH RX INTERRUPTS DISABLED."
 035300                                    ERRDF   2608,EM2606,ER0504;          >>>>> ERROR #2608 <<<<<.
 035300  104455                                                                  TRAP    C$ERDF
 035302  005060                                                                  .WORD   2608
 035304  013467                                                                  .WORD   EM2606
 035306  016174                                                                  .WORD   ER0504
                                   ;+
                                   ; DISABLE INTERRUPTS AND CLEAN OUT THE INTERRUPT VECTORS USED IN THIS TEST.
                                   ;-
 035310  005001                    32$:    CLR     R1               ;CLEAR BOTH TRANSMITTER
 035312  004767  165456                    JSR     PC,TXIEO         ; INTERRUPT ENABLE AND RECEIVER
 035316  004767  164670                    JSR     PC,RXIEO         ; INTERRUPT ENABLE BITS IN THE OUT CSR.
 035322                            34$:    SETPRI  #PRI06           ;DISABLE DEVICE INTERRUPTS.
 035322  012700  000300                                                          MOV     #PRI06,R0
 035326  104441                                                                  TRAP    C$SPRI
 035330                                    CLRVEC  RXVECA           ;RETURN RX INT VECTOR TO UNUSED POOL.
 035330  016700  144676                                                          MOV     RXVECA,R0
 035334  104436                                                                  TRAP    C$CVEC
 035336                                    CLRVEC  TXVECA           ;RETURN TX INT VECTOR TO UNUSED POOL.
 035336  016700  144672                                                          MOV     TXVECA,R0
 035342  104436                                                                  TRAP    C$CVEC

 035344  005067  144736            60$:    CLR     CTRLCF           ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 035350                                    SETPRI  #PRI07           ;DISABLE ALL INTERRUPTS.
 035350  012700  000340                                                          MOV     #PRI07,R0
 035354  104441                                                                  TRAP    C$SPRI
 035356                                    ENDTST
 035356                                                                  L10054:
 035356  104401                                                                  TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE TEST           - BRLEVA -
                                        ;++ ********************************************************************
                                        ;*                      - BR LEVEL TEST 8 -
                                        ;*      THIS TEST VERIFIES THAT THE DEVICE UNDER TEST (DUT) WILL GENERATE
                                        ;*      RECEPTION AND TRANSMISSION INTERRUPTS AT THE CORRECT BR LEVEL.
                                        ;*      THIS TEST DOES NOT DEPEND ON THE USE OF THE SERIAL LINE TRANSMISSION
                                        ;*      OR RECEPTION CAPABILITIES OF THE DUT.  THE LINES ARE PUT IN INTERNAL
                                        ;*      LOOPBACK TO MINIMIZE ANY EXTERNAL EFFECTS THAT COULD BE CAUSED ON
                                        ;*      DEVICES ATTACHED TO THE SERIAL LINES.
                                        ;*
                                        ;-- ********************************************************************

 035360                                         BGNTST
 035360                                                                                  T24::
 035360                                         SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
 035360  012700  000240                                                                          MOV     #PRIO5,R0
 035364  104441                                                                                  TRAP    C$SPRI
         000030                                  TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 035366  012767  000030  144730                  MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (30)
 035374  012767  177777  144704                  MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
 035402  012767  000001  146356                  MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 035410  012767  005671  146352                  MOV     #3001.,ERRNBR  ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
 035416  012767  014223  146346                  MOV     #EM3001,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
 035424  005067  145032                          CLR     ERSMRF         ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
                                        ;+
                                        ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
                                        ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                        ; THIS SUBROUTINE REPORTS ERRORS FROM >>>>> 3001 THRU 3002 <<<<<.
                                        ;-
 035430  004767  163726                          JSR     PC,RESETT      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
 035434  103402                                  BCS     2$             ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
 035436  000167  000644                          JMP     60$            ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
 035442  012767  005673  146320  2$:             MOV     #3003.,ERRNBR  ;SET THE ERROR REPORT NUMBER TO 3003.
                                        ;+
                                        ; ENABLE TRANSMITTERS ON ALL LINES.
                                        ;-
 035450  012705  177777          4$:             MOV     #MAPLNS,R5     ;PASS ACTIVE LINE BIT MAP.
 035454  004767  165220                          JSR     PC,TXENBL      ;ENABLE TRANSMISSION ON ALL LINES.
                                        ;+
                                        ; GENERATE A TRANSMISSION INTERRUPT REQUEST.
                                        ; PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING INTS.
                                        ;-
 035460                                         SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
 035460  012700  000340                                                                          MOV     #PRIO7,R0
 035464  104441                                                                                  TRAP    C$SPRI
 035466                                         SETVEC  TXVECA,#TXINTR,#PRIO7 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
 035466  012746  000340                                                                          MOV     #PRIO7,-(SP)
 035472  012746  024146                                                                          MOV     #TXINTR,-(SP)
 035476  016746  144532                                                                          MOV     TXVECA,-(SP)
 035502  012746  000003                                                                          MOV     #3,-(SP)
 035506  104437                                                                                  TRAP    C$SVEC
 035510  062706  000010                                                                          ADD     #10,SP
                                                ;+
                                                ; SET UP DUT FOR TRANSMISSION INTERRUPTS:
                                                ;  SET UP INTERNAL LOOPBACK.
                                                ;  SET UP LINE PARAMETERS FOR TRANSMISSION.
                                                ;
```

```
                        ;+
                        ; SEND A NULL CHAR TO EACH LINE.
                        ;-
7116 035540  016701  144544          MOV     IESTAT,R1       ;SET UP THE STATE OF THE INTERRUPT ENABLE BITS.
7117 035544  010177  144472    5$:   MOV     R1,@CSRA        ;SET UP THE LINE NUMBER AND INTERRUPT ENABLE
7118                                                         ;BITS IN THE CSR.
7119 035550  112777  000000  144472  MOVB    #0,@TDATA       ;SEND A NULL CHARACTER TO THE OUTPUT FIFO.
7120 035556  005201                  INC     R1              ;NEXT LINE.
7121 035560  020127  000020          CMP     R1,#NUMLNS      ;IF ALL LINES HAVE NOT BEEN SERVICED THEN
7122 035564  002767                  BLT     5$              ;BRANCH.
                        ;+
                        ; DELAY 50 MS TO ALLOW TIME FOR THE INTERRUPT TO BE GENERATED.
                        ;-
7126 035566  012704  000062          MOV     #50.,R4         ;PASS 50 MS TIME TO THE DELAY ROUTINE.
7127 035572  004767  162326          JSR     PC,DELAY        ;DELAY 50 MILLI-SECONDS.
                        ;+
                        ; GENERATE A RECEPTION INTERRUPT REQUEST.
                        ;-
7131 035576                          SETVEC  RXVECA,#RXBRRT,#PRIO7 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
     035576  012746  000340                                                  MOV     #PRIO7,-(SP)
     035602  012746  023756                                                  MOV     #RXBRRT,-(SP)
     035606  016746  144420                                                  MOV     RXVECA,-(SP)
     035612  012746  000003                                                  MOV     #3,-(SP)
     035616  104437                                                          TRAP    C$SVEC
     035620  062706  000010                                                  ADD     #10,SP
                        ;+
                        ; SET UP FOR THE LOOP WHICH TESTS THE INTERRUPT BR LEVELS.
                        ;-
7155 035624  012705  000340          MOV     #340,R5         ;SET UP THE PRIORITY LEVEL TO 7.
7156 035630  005003                  CLR     R3              ;CLEAR THE RX PRIORITY STORE AND FLAGS.
7157 035632  005002                  CLR     R2              ;CLEAR THE TX PRIORITY STORE AND FLAGS.
                        ;+
                        ; ENABLE TX AND RX INTERRUPTS.
                        ; PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING THE INTERRUPTS.
                        ;-
7161 035634  004767  164412          JSR     PC,RXIE1        ;ENABLE RECEIVER INTERRUPTS.
7162 035640  004767  165170          JSR     PC,TXIE1        ;ENABLE TRANSMITTER INTERRUPTS.
                        ;+
                        ; LOOP, LOWERING THE PROCESSOR PRIORITY UNTIL THE DUT INTERRUPTS ON RX AND TX.
                        ;-
7166 035644  005067  144456  6$:   CLR     TXINTC          ;CLEAR THE TX INTERRUPT COUNTER.
7167 035650  005067  144454          CLR     TXINTF          ;CLEAR THE TX INTERRUPT FLAGS.
7168 035654  005067  144434          CLR     RXINTC          ;CLEAR THE RX INTERRUPT COUNTER.
7169 035660  005067  144432          CLR     RXINTF          ;CLEAR THE RX INTERRUPT FLAGS.
7170 035664                          SETPRI  R5              ;SET PROCESSOR PRIORITY TO THE SELECTED VALUE.
     035664  010500                                                          MOV     R5,R0
     035666  104441                                                          TRAP    C$SPRI
7172 035670  012704  000001          MOV     #1,R4           ;PASS 1 MS COUNT TO THE DELAY ROUTINE.
7173 035674  004767  162224          JSR     PC,DELAY        ;DELAY 1 MS TO ALLOW INTERRUPTS TO OCCUR.
                        ;+
                        ; DETERMINE IF ANY RX DUT INTERRUPTS OCCURRED.
                        ; LOG THE PROCESSOR PRIORITY FOR THE RX INTERRUPT IF FIRST RX INT.
```

```
7157                                    ;-
7158 035700  005767  144410                 TST     RXINTC          ;CHECK THE RECEIVE INTERRUPT COUNTER.
7159 035704  001412                         BEQ     8$              ;SKIP THE PRIORITY LOG IF NO RX INT OCCURRED.
7160                                        ;+
7161                                        ; IF THIS IS THE FIRST RX INTERRUPT, LOG THE PRIORITY.
7162                                        ;
7163 035706  005703                         TST     R3              ;CHECK THE RX PRIORITY STORE AND FLAGS.
7164 035710  001010                         BNE     8$              ;GOTO TEST FOR TX INTS IF NOT THE FIRST RX INT.
7165 035712  010503                         MOV     R5,R3           ;LOG THE PRESENT PRIORITY IN THE RX PRIO STORE.
7166 035714  052703  100000                 BIS     #BIT15,R3       ;SET THE RX INT HAS OCCURRED FLAG.
7167 035720  016700  144372                 MOV     RXINTF,R0       ;GET THE RX INTERRUPT ROUTINE FLAGS.
7168 035724  042700  137777                 BIC     #137777,R0      ;CLEAR ALL BUT THE TX INT ERROR FLAG.
7169 035730  050003                         BIS     R0,R3           ;IF TX INT ERROR, SET BIT 14 OF THE PRIO FLAGS.
7170                                    ;+
7171                                    ; DETERMINE IF ANY TX OUT INTERRUPTS HAVE OCCURRED.
7172                                    ; LOG THE PRESENT PROCESSOR PRIORITY IF THIS IS THE FIRST TX INTERRUPT.
7173                                    ;-
7174 035732  005767  144370         8$:     TST     TXINTC          ;CHECK THE TRANSMIT INTERRUPT COUNTER.
7175 035736  001405                         BEQ     10$             ;SKIP THE PRIORITY LOG IF NO TX INT OCCURRED.
7176                                        ;+
7177                                        ; IF THIS IS THE FIRST TX INTERRUPT, LOG THE PRIORITY.
7178                                        ;
7179 035740  005702                         TST     R2              ;CHECK THE TX PRIORITY STORE AND FLAGS.
7180 035742  100403                         BMI     10$             ;SKIP THE LOGGING IF NOT FIRST TX INTERRUPT.
7181 035744  010502                         MOV     R5,R2           ;LOG THE PRESENT PRIORITY IN THE TX PRIO STORE.
7182 035746  052702  100000                 BIS     #BIT15,R2       ;SET THE TX INT HAS OCCURRED FLAG.
7183                                    ;+
7184                                    ; SELECT NEXT PROCESSOR PRIORITY.
7185                                    ; TEST FOR BOTH RX AND TX INTERRUPTS HAVING OCCURRED, LOOP IF NOT.
7186                                    ;-
7187 035752  162705  000040         10$:    SUB     #40,R5          ;DECREMENT PRIORITY LEVEL BY ONE.
7188 035756  002402                         BLT     12$             ;GOTO CHECK FOR ERRORS IF BELOW PRIORITY ZERO.
7189 035760  030305                         BIT     R3,R3           ;AND PRIO FLAGS TOGETHER, ALTER NONE OF THEM.
7190 035762  100330                         BPL     6$              ;LOOP IF RX AND TX INTS HAVEN'T BOTH OCCURRED.
7191
7192                                    ;+
7193                                    ; DISABLE INTERRUPTS AND CLEAR INTERRUPT VECTORS.
7194 035764                             ;
      035764  012700  000340         12$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
                                                                                MOV     #PRI07,R0
      035770  104441                                                            TRAP    C$SPRI
7195 035772                                 CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
      035772  016700  144234                                                    MOV     RXVECA,R0
      035776  104437                                                            TRAP    C$CVEC
7196 036000                                 CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
      036000  016700  144230                                                    MOV     TXVECA,R0
      036004  104436                                                            TRAP    C$CVEC
7197                                    ;+
7198                                    ; VERIFY THAT RX AND TX INTERRUPTS OCCURRED,
7199                                    ; AT THE PROPER BR LEVEL, AND
7200                                    ; IN THE PROPER ORDER.
7201                                    ;    DETERMINE IF TX INTERRUPT OCCURRED.
7202                                    ;-
7203 036006  005702                         TST     R2              ;DETERMINE WHETHER TX INT OCCURRED OR NOT.
7204 036010  100420                         BMI     16$             ;SKIP THESE ERRORS IF TX INT OCCURRED.
7205                                    ;+
7206                                    ; DETERMINE REASON THAT NO TX INT OCCURRED.
7207                                    ;-
```

```
036012  012701  013752                  MOV     #EM2610,R1       ;SELECT "NO TX INT FROM TX.ACTION" MESSAGE.
036016  005777  144220                  TST     @CSRA            ;CHECK THE TX.ACTION BIT OF THE DUT CSR.
036022  100402                          BMI     14$              ;SKIP TX.ACTION CLR MSG SELECTION IF IT IS SET.
036024  012701  014044                  MOV     #EM2611,R1       ;SELECT "TX.ACTION CLEAR AFTER CHARS SENT" MSG.
                                ;REPORT "INTERRUPT BR LEVEL TEST ERROR:"
036030                          14$:    ERRDF   3003,EM3001,ER0503;          >>>>> ERROR #3003 <<<<<.
036030  104455                                                                    TRAP    C$ERDF
036032  005673                                                                    .WORD   3003
036034  014223                                                                    .WORD   EM3001
036036  016156                                                                    .WORD   ER0503

                                ;*
                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                ;-
036040  032767  000100  144156          BIT     #BIT06,OPTION    ;EXIT WITH TEST FAILURE MESSAGE IF
036046  001513                          BEQ     26$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED

036050  000427                          BR      18$              ;SKIP THE BR LEVEL CHECK, NO TX INT OCCURRED.
                                ;*
                                ; VERIFY THAT THE TX INTERRUPT WAS AT THE PROPER BR LEVEL.
                                ;-
036052  010204                  16$:    MOV     R2,R4            ;CALCULATE THE BR LEVEL
036054  042704  177400                  BIC     #177400,R4       ; THAT THE TRANMSIT
036060  006204                          ASR     R4               ; INTERRUPT WAS
036062  006204                          ASR     R4               ; REQUESTED AT, WHICH
036064  006204                          ASR     R4               ; IS ONE GREATER THAN
036066  006204                          ASR     R4               ; THE PROCESSOR PRIORITY
036070  006204                          ASR     R4               ; LEVEL AT WHICH THE
036072  005204                          INC     R4               ; TRANSMIT INTERRUPT OCCURRED.
036074  116705  144140                  MOVB    BRLEVL,R5        ;GET THE EXPECTED INTERRUPT BR LEVEL.
036100  120405                          CMPB    R4,R5            ;COMPARE THE INTERRUPT BR LEVEL WITH EXPECTED.
036102  001412                          BEQ     18$              ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
                                ;REPORT "TX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
036104  012701  014356                  MOV     #EM3003,R1       ;SELECT THE ERROR MESSAGE FOR THE ERROR CALL.
036110                                  ERRDF   3004,EM3001,ER3001;          >>>>> ERROR #3004 <<<<<.
036110  104455                                                                    TRAP    C$ERDF
036112  005674                                                                    .WORD   3004
036114  014223                                                                    .WORD   EM3001
036116  016620                                                                    .WORD   ER3001
                                ;*
                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                ;-
036120  032767  000100  144076          BIT     #BIT06,OPTION    ;EXIT WITH TEST FAILURE MESSAGE IF
036126  001463                          BEQ     26$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED

                                ;*
                                ; DETERMINE IF RX INTERRUPT OCCURRED.
                                ;
036130  005705                  18$:    TST     R5               ;CHECK THE RX INT OCCURRED FLAG.
036132  100421                          BMI     22$              ;SKIP THESE ERRORS IF RX INT OCCURRED.
                                ;*
                                ; DETERMINE REASON THAT NO RX INT OCCURRED.
                                ;
036134  012701  014127                  MOV     #EM2612,R1       ;SELECT "NO RX INT FROM RX.DATA.AVAIL" MSG.
036140  032777  000200  144074          BIT     #BIT7,@CSRA      ;CHECK THE RX.DATA.AVAIL BIT OF THE DUT CSR.
036146  001002                          BNE     20$              ;SKIP RX.DATA.AVAIL CLR MSG IF BIT IS SET.
036150  012701  014262                  MOV     #EM3002,R1       ;SELECT "NO RX.DATA.AVAIL AFTER RESET" MSG.
```

```
7257                                    ;REPORT "INTERRUPT BR LEVEL TEST ERROR:"
7258 036154                         20$:    ERRDF   3005,EM3001,ERO503;          >>>>> ERROR #3005 <<<<<.
     036154  104455                                                                       TRAP    C$ERDF
     036156  005675                                                                       .WORD   3005
     036160  014223                                                                       .WORD   EM3001
     036162  016136                                                                       .WORD   ERO503
7259                                    ;*
7260                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7261                                    ;-
7262 036164  032767  000100  144032        BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
7263 036172  001441                       BEQ     26$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7264
7265 036174  000427                       BR      24$             ;SKIP THE BR CHECK IF NO RX INT OCCURRED.
7266                                    ;*
7267                                    ; VERIFY THAT THE RX INTERRUPT WAS AT THE PROPER BR LEVEL.
7268                                    ;-
7269 036176  010304                 22$:    MOV     R3,R4           ;CALCULATE THE BR LEVEL
7270 036200  042704  177400                BIC     #177400,R4      ; THAT THE RECEIVE
7271 036204  006204                        ASR     R4              ; INTERRUPT WAS
7272 036206  006204                        ASR     R4              ; REQUESTED AT, WHICH
7273 036210  006204                        ASR     R4              ; IS ONE GREATER THAN
7274 036212  006204                        ASR     R4              ; THE PROCESSOR PRIORITY
7275 036214  006204                        ASR     R4              ; LEVEL AT WHICH THE
7276 036216  005204                        INC     R4              ; RECEIVE INTERRUPT OCCURRED.
7277 036220  116705  144014                MOVB    BRLEVL,R5       ;GET THE EXPECTED INTERRUPT BR LEVEL.
7278 036224  120405                       CMPB    R4,R5           ;COMARE THE INTERRUPT BR LEVEL WITH EXPECTED.
7279 036226  001412                       BEQ     24$             ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
7280                                    ;REPORT "RX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
7281 036230  012701  014432                MOV     #EM3004,R1      ;SELECT ERROR MESSAGE FOR THE ERROR CALL.
7282 036234                             ERRDF   3006,EM3001,ER3001;          >>>>> ERROR #3006 <<<<<.
     036234  104455                                                                       TRAP    C$ERDF
     036236  005676                                                                       .WORD   3006
     036240  014223                                                                       .WORD   EM3001
     036242  016620                                                                       .WORD   ER3001
7283                                    ;*
7284                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7285                                    ;-
7286 036244  032767  000100  143752        BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
7287 036252  001411                       BEQ     26$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7288
7289                                    ;*
7290                                    ; TEST FOR INTERRUPTS OCCURING IN THE PROPER ORDER.
7291                                    ;-
7292 036254  032705  040000         24$:    BIT     #BIT14,R5       ;CHECK THE IMPROPER INT ORDER ERROR FLAG.
7293 036260  001406                       BEQ     26$             ;SKIP ERROR REPORT IF ERROR DID NOT OCCUR.
7294                                    ;REPORT "TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS RX INT."
7295 036262  012701  014506                MOV     #EM3005,R1      ;SELECT THE ERROR MESSAGE FOR INDIRECT PRINT.
7296 036266                             ERRDF   3007,EM3001,ER0503;          >>>>> ERROR #3007 <<<<<.
     036266  104455                                                                       TRAP    C$ERDF
     036270  005677                                                                       .WORD   3007
     036272  014223                                                                       .WORD   EM3001
     036274  016136                                                                       .WORD   ER0503
7297                                    ;*
7298                                    ; CLEAN UP, EXIT THE TEST.
7299                                    ;
7300 036276  004767  164476         26$:    JSR     PC,TXIEO        ;CLEAR TRANSMITTER INTERRUPTS.
7301 036302  004767  163704                JSR     PC,RXIEO        ;CLEAR RECEIVER INTERRUPTS.
```

```
 7302 036306  005067  143774          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7303 036312                          SETPRI  @PRIO7          ;DISABLE ALL INTERRUPTS.
      036312  012700  000340                                                  MOV     @PRIO7,R0
      036316  104441                                                         TRAP    C$SPRI
 7304 036320                          ENDTST
      036320                                                          L10055:
      036320  104401                                                         TRAP    C$ETST
 7305
```

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

                                        BGNTST
                                                                                        T25::
                                        SETPRI  #PRIOS          ;ALLOW LTC INTERRUPTS.
                                                                                        MOV     #PRIOS,R0
                                                                                        TRAP    C$SPRI
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
                                         MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (31)
                                         MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
                                         MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
                                         MOV    #3101.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
                                         MOV    #EM3101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
                                         MOV    #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 3101 <<<<<.
                                ;-
                                        JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
                                        BCC     60$             ;RESET FAILURE?, ABORT THIS TEST.
                                ;+
                                ; TEST ALL ACTIVE LINES INDIVIDUALLY.
                                ; WRITE THE REQUEST CODE TO THE DIAGNOSTIC FIELD IN THE LPR REGISTER.
                                ; VERIFY THAT A BMP CODE IS RETURNED WITHIN THE CORRECT TIME.
                                ;-
                                          MOV   ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
                                          CLR   R4              ;CLEAR THE LINE NUMBER COUNTER.
                                          MOV   CSRA,R3         ;GET THE ADDRESS OF THE DUT'S CSR.
                                2$:       CLC                   ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
                                          ROR   R5              ;SHIFT THE BIT MAP INTO THE CARRY BIT.
                                         BCC    9$             ;DO NOT TEST THE LINE IF IT IS INACTIVE.
                                    ;+
                                    ; SELECT THE LINE UNDER TEST.
                                    ;  WRITE THE BMP REQUEST CODE TO THE DIAG FIELD IN THE LPR REGISTER.
                                    ;-
                                      MOV   #3102.,ERRNBR   ;SET THE ERROR NUMBER TO 3102.
                                      MOV   R4,(R3)         ;SELECT THE LINE CURRENTLY UNDER TEST.
                                      BIS   #2,@LPRA        ;WRITE THE BMP REQUEST CODE TO THE LPR.
                                    ;+
                                    ; WAIT FOR BMP REQUEST CODE TO BE CLEARED, REPORT ERROR IF TIME-OUT
                                    ; OCCURS.
                                    ;
                                      MOV   #11750,R1       ;TEST BIT 1, TIMEOUT OF 1 SEC.
                                      MOV   LPRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
                                      JSR   PC,WAIBIC       ;WAIT FOR REQUEST CODE TO CLEAR.
                                     BCC    6$             ;GO REPORT ERROR IF CODE DID NOT CLEAR IN TIME.
                                    ;+
                                    ; WAIT FOR BMP CODE TO APPEAR IN THE FIFO, REPORT ERROR IF TIME OUT
                                    ; OCCURS.
                                    ;-
```

```
 7361 036456  005267  145306                  INC     ERRNBR          ;SET ERROR NUMBER TO 3103.
 7362 036462  012701  070012                  MOV     #70012,R1       ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
 7363 036466  016702  143550                  MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
 7364 036472  004767  164572                 JSR     PC,WAIBIS       ;WAIT FOR RX_DATA_AVAILABLE TO SET.
 7365 036476  103031                        BCC     6$              ;GO REPORT ERROR IF CODE DID NOT CLEAR IN TIME.
 7366                                       ;+
 7367                                       ; READ THE BMP CODE (IF IT IS THERE) FROM THE RBUF REGISTER.
 7368                                       ; DETERMINE IF IT IS A VALID BMP CODE.
 7369                                       ; VERIFY THE BMP CODE WAS RECEIVED FROM THE CORRECT CHANNEL.
 7370                                       ; IF THE BMP CODE DOES NOT INDICATE DUT RUNNING OK, THEN SAVE IT ON
 7371                                       ; THE QUEUE TO BE REPORTED IN A LATER TEST.
 7372                                       ;-
 7373 036500  005267  145264                  INC     ERRNBR          ;SET ERROR NUMBER TO 3104.
 7374 036504  017702  143534                  MOV     @RBUFA,R2       ;GET THE BMP CODE FROM THE FIFO.
 7375 036510  100024                        BPL     6$              ;GO REPORT ERROR IF NO BMP CODE FOUND.
 7376 036512  005267  145252                  INC     ERRNBR          ;SET ERROR NUMBER TO 3105.
 7377 036516  012700  170301                  MOV     #170301,R0      ;SET-UP A BMP CODE MASK.
 7378 036522  040200                         BIC     R2,R0           ;TRY TO CLEAR THE BMP MASK.
 7379 036524  001016                        BNE     6$              ;GO REPORT ERROR IF IT IS NOT A VALID BMP CODE.
 7380 036526  005267  145236                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 3106.
 7381 036532  010200                         MOV     R2,R0           ;COPY THE BMP CODE.
 7382 036534  000300                         SWAB    R0              ;PUT THE LINE NUMBER IN THE LOW BYTE.
 7383 036536  042700  177760                  BIC     #177760,R0      ;CLEAR THE UNWANTED BITS.
 7384 036542  120400                         CMPB    R4,R0           ;DID THE BMP CODE COME FROM THE CORRECT LINE?.
 7385 036544  001006                        BNE     6$              ;NO; GO REPORT ERROR.
 7386 036546  120227  000305                  CMPB    R2,#305         ;IS THE BMP CODE A "GOOD ONE"?.
 7387 036552  001413                        BEQ     8$              ;YES; SKIP SAVING THE BMP CODE ON THE QUEUE.
 7388 036554  004767  163516                 JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
 7389 036560  000410                        BR      8$              ;GO SEE IF THERE ARE ANY MORE LINE TO TEST.
 7390
 7391 036562  010401                 6$:     MOV     R4,R1           ;PASS THE LINE NUMBER TO BE REPORTED.
 7392 036564  012702  014643                 MOV     #EM3102,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
 7393                                                                ;"BMP REQUEST BIT BAD ON LINE:"
 7394 036570                                 ERROR                   ;               >>>>> ERROR <<<<<.
      036570  104460                                                                       TRAP    C$ERROR
 7395
 7396                                       ;+
 7397                                       ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
 7398                                       ;-
 7399 036572  032767  000100  143424            BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
 7400 036600  001403                        BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
 7401
 7402                                       ;+
 7403                                       ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
 7404                                       ;-
 7405 036602  005204                 8$:     INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
 7406 036604  005705                         TST     R5              ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
 7407 036606  001302                        BNE     .3              ;YES; BRANCH TO TEST THE NEXT LINE.
 7408 036610  005067  143472            60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7409 036614                                 ENDTST
      036614                                                                    L10056:
      036614  104401                                                                TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE TEST            - REPBMP -
                                        ;++ *************************************************************************
                                        ;*                          REPORT ANY BMP CODES IN THE QUEUE
                                        ;*      THIS IS A PSEUDO-TEST USED TO REPORT ANY BMP CODES THAT WERE FOUND
                                        ;*      IN THE DUT'S FIFO DURING PREVIOUS TEST, AND LOGGED IN THE BMP CODE
                                        ;*      QUEUE.
                                        ;*      IT IS UNLIKELY THAT RUNNING THIS PSEUDO-TEST ALONE WILL PRODUCE ANY
                                        ;*      ERROR REPORTS.
                                        ;*
                                        ;-- *************************************************************************
036616                                          BGNTST
036616                                                                          T26::
        000032                                   TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
036616  012767  000032  143500                   MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (93)
036624  012767  177777  143454                   MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
036632  016702  143666                           MOV     BMPCQP R2       ;GET THE CONTENTS OF THE POINTER.
036636  012703  002526                           MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
036642  020203                                   CMP     R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
036644  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"

036646  012701  015433                          MOV     #EM9304,R1      ;PASS THE FIRST MESSAGE TO BE REPORTED.
036652                                          ERRDF   9301,EM9301,ER9301 ;           >>>>> ERROR #9301 <<<<<.
036652  104455                                                                                  TRAP    C$ERDF
036654  022125                                                                                  .WORD   9301
036656  015257                                                                                  .WORD   EM9301
036660  017262                                                                                  .WORD   ER9301

036662  012767  002526  143634                   MOV     #BMPCQB,BMPCQP  ;SET POINTER BACK TO THE BEGINING OF THE QUE.

036670  005067  143412                  60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
036674                                          ENDTST
036674                                                                          L10057:
036674  104401                                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE PARAMETER CODING SECTION



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

036676                                  BGNHRD
036676  000022                                                          .WORD L10060-L$HARD/2
036700                                                          L$HARD::

                                ;DEVICE CSR ADDRESS QUESTION:
036700                                  GPRMA   HWPTQ1,0,0,160000,177776,YES
036700  000031                                                          .WORD   T$CODE
036702  036744                                                          .WORD   HWPTQ1
036704  160000                                                          .WORD   T$LOLIM
036706  177776                                                          .WORD   T$HILIM

                                ;DEVICE INTERRUPT VECTOR QUESTION:
036710                                  GPRMA   HWPTQ2,2,0,40,776,YES
036710  001051                                                          .WORD   T$CODE
036712  036762                                                          .WORD   HWPTQ2
036714  000040                                                          .WORD   T$LOLIM
036716  000776                                                          .WORD   T$HILIM

                                ;ACTIVE LINES BIT MAP QUESTION:
036720                                  GPRMD   HWPTQ3,4,0,MAPLNS,0,MAPLNS,YES
036720  002032                                                          .WORD   T$CODE
036722  037015                                                          .WORD   HWPTQ3
036724  177777                                                          .WORD   MAPLNS
036726  000000                                                          .WORD   T$LOLIM
036730  177777                                                          .WORD   T$HILIM

                                ;INTERRUPT BR LEVEL QUESTION:
036732                                  GPRMD   HWPTQ4,6,0,377,0,6,YES
036732  003032                                                          .WORD   T$CODE
036734  037043                                                          .WORD   HWPTQ4
036736  000377                                                          .WORD   377
036740  000000                                                          .WORD   T$LOLIM
036742  000006                                                          .WORD   T$HILIM


036744                                  ENDHRD
                                                                        .EVEN
036744                                                          L10060:


036744  103     123     123     123     HWPTQ1: .ASCIZ  /CSR ADDRESS: /
036747  040     101     104
036752  104     122     105
036755  123     123     072
036760  040     000
```

```
7496 036762  111  116  124      HWPTQ2: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
     036765  105  122  122
     036770  125  120  124
     036773  040  126  105
     036776  103  124  117
     037001  122  040  101
     037004  104  104  122
     037007  105  123  123
     037012  072  040  000
7497 037015  101  103  124      HWPTQ3: .ASCIZ  /ACTIVE LINE BIT MAP: /
     037020  111  126  105
     037023  040  114  111
     037026  116  105  040
     037031  102  111  124
     037034  040  115  101
     037037  120  072  040
     037042  000
7498 037043  111  116  124      HWPTQ4: .ASCIZ  /INTERRUPT BR LEVEL: /
     037046  105  122  122
     037051  125  120  124
     037054  040  102  122
     037057  040  114  105
     037062  126  105  114
     037065  072  040  000
7499
7500                            .EVEN
```

```
                                        .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.
                                        ;--

     037070                                     BGNSFT
     037070  000017                                                                              .WORD L10061-L$SOFT/2
     037072                                                                              L$SOFT::

                                        ;UNIT NUMBER PRINTOUT QUESTION:
     037072                                     GPRML   SWPTQ1,0,20,YES
     037072  000130                                                                              .WORD   T$CODE
     037074  037130                                                                              .WORD   SWPTQ1
     037076  000020                                                                              .WORD   20
                                        ;ROM VERSION NUMBER PRINTOUT ON FIRST PASS QUESTION:
     037100                                     GPRML   SWPTQ2,0,1,YES
     037100  000130                                                                              .WORD   T$CODE
     037102  037204                                                                              .WORD   SWPTQ2
     037104  000001                                                                              .WORD   1
                                        ;EXTENDED ERROR REPORTING QUESTION:
     037106                                     GPRML   SWPTQ3,0,100,YES
     037106  000130                                                                              .WORD   T$CODE
     037110  037255                                                                              .WORD   SWPTQ3
     037112  000100                                                                              .WORD   100
                                          ;+
                                          ; IF EXTENDED ERROR REPORTING IS NOT REQUIRED THEN SKIP THE NEXT QUESTION.
                                          ;+
     037114                                     XFERF   ENDD
     037114  006044                                                                              .WORD   T$CODE

                                        ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
     037116                                     GPRMD   SWPTQ4,2,D,177777,0,177777,YES
     037116  001052                                                                              .WORD   T$CODE
     037120  037310                                                                              .WORD   SWPTQ4
     037122  177777                                                                              .WORD   177777
     037124  000000                                                                              .WORD   T$LOLIM
     037126  177777                                                                              .WORD   T$HILIM
                                                .EVEN

     037130                             ENDD:   ENDSFT
                                                                                                 .EVEN
     037130                                                                              L10061:


     037130     122     105     120             SWPTQ1: .ASCIZ  /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /
     037133     117     122     124
     037136     040     125     116
     037141     111     124     040
     037144     116     125     115
```

```
                037147  102     105     122
                037152  040     101     123
                037155  040     105     101
                037160  103     110     040
                037163  125     116     111
                037166  124     040     111
                037171  123     040     124
                037174  105     123     124
                037177  105     104     072
                037202  040     000
7557            037204  122     117     115             SWPTQ2: .ASCIZ  /ROM VERSION PRINTOUT ON THE FIRST PASS: /
                037207  040     126     105
                037212  122     123     111
                037215  117     116     040
                037220  120     122     111
                037223  116     124     117
                037226  125     124     040
                037231  117     116     040
                037234  124     110     105
                037237  040     106     111
                037242  122     123     124
                037245  040     120     101
                037250  123     123     072
                037253  040     000
7558            037255  105     130     124             SWPTQ3: .ASCIZ  /EXTENDED ERROR REPORTING: /
                037260  105     116     104
                037263  105     104     040
                037266  105     122     122
                037271  117     122     040
                037274  122     105     120
                037277  117     122     124
                037302  111     116     107
                037305  072     040     000
7559            037310  116     125     115             SWPTQ4: .ASCIZ  /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /
                037313  102     105     122
                037316  040     117     106
                037321  040     111     116
                037324  104     111     126
                037327  111     104     125
                037332  101     114     040
                037335  104     101     124
                037340  101     040     105
                037343  122     122     117
                037346  122     123     040
                037351  124     117     040
                037354  122     105     120
                037357  117     122     124
                037362  040     117     116
                037365  040     101     040
                037370  114     111     116
                037373  105     072     040
                037376  000
7560                                                    .EVEN
```

```
7569
7570
7571 037400                          $PATCH::
7572 037400                                  .BLKW   24
7573
7580
7581
7582
7583
7584 037450                                  LASTAD
                                                                .EVEN
     037450  000000                                           .WORD   0
     037452  000000                                           .WORD   0
     037454                          L$LAST::
7585 037454                                  ENDMOD
7586
7587
7588
7589
7590
7591
7592
7593         000001                          .END
```

SYMBOL TABLE

```
ACTLNS  000230 G      C$AU    = 000052      DROP    025122      EM0902  010766 G      EM9023  015217 G
ADR     = 000020 G    C$AUTO  = 000061      DR00MG  005464 G    EM1001  011022 G      EM9024  015241 G
ADRPTR  017764 G      C$BRK   = 000022      DR02MG  005470 G    EM1002  011055 G      EM9301  015257 G
ALTFLD  017454 G      C$BSEG  = 000004      DR04MG  005475 G    EM1003  011142 G      EM9302  015336 G
ASSEMB  = 000010      C$BSUB  = 000002      DR06MG  005501 G    EM1101  011220 G      EM9303  015366 G
BCOUNT  002352 G      C$CEFG  = 000045      DR10MG  005517 G    EM1201  011273 G      EM9304  015433 G
BITTBL  002410 G      C$CLCK  = 000062      DR12MG  005526 G    EM1202  011321 G      ENDD    037130
BIT0    = 000001 G    C$CLEA  = 000012      DR14MG  005537 G    EM1203  011405 G      ENDFTB  003726 G
BIT00   = 000001 G    C$CLOS  = 000035      DR16MG  005550 G    EM1204  011463 G      ENDIT   025042
BIT01   = 000002 G    C$CLP1  = 000006      EDROP   025200      EM1205  011517 G      ERCNTB  002464 G
BIT02   = 000004 G    C$CVEC  = 000036      EF.CON  = 000036 G  EM1301  011543 G      ERLTBL  002726 G
BIT03   = 000010 G    C$DCLN  = 000044      EF.NEW  = 000035 G  EM1302  011576 G      ERRBLK  003774 G
BIT04   = 000020 G    C$DODU  = 000051      EF.PWR  = 000034 G  EM1401  011635 G      ERRMSG  003772 G
BIT05   = 000040 G    C$DRPT  = 000024      EF.RES  = 000037 G  EM1402  011674 G      ERRNBR  003770 G
BIT06   = 000100 G    C$DU    = 000053      EF.STA  = 000040 G  EM1403  011762 G      ERRTYP  003766 G
BIT07   = 000200 G    C$EDIT  = 000003      EF0503  004074 G    EM1404  012035 G      ERSMRF  002462 G
BIT08   = 000400 G    C$ERDF  = 000055      EF0505  004101 G    EM1405  012107 G      ER0101  015510 G
BIT09   = 001000 G    C$ERHR  = 000056      EF1401  004154 G    EM1406  012122 G      ER0301  016042 G
BIT1    = 000002 G    C$ERRO  = 000060      EF1402  004256 G    EM1407  012135 G      ER0503  016136 G
BIT10   = 002000 G    C$ERSF  = 000054      EF1601  004313 G    EM1408  012147 G      ER0504  016174 G
BIT11   = 004000 G    C$ERSO  = 000057      EF1602  004345 G    EM1501  012155 G      ER1401  016254 G
BIT12   = 010000 G    C$ESCA  = 000010      EF1603  004407 G    EM1502  012203 G      ER1601  016416 G
BIT13   = 020000 G    C$ESEG  = 000005      EF1604  004451 G    EM1601  012221 G      ER1603  016526 G
BIT14   = 040000 G    C$ESUB  = 000003      EF3001  004546 G    EM1604  012304 G      ER3001  016620 G
BIT15   = 100000 G    C$ETST  = 000001      EF3002  004615 G    EM1701  012360 G      ER9004  016722 G
BIT2    = 000004 G    C$EXIT  = 000032      EF9006  004664 G    EM1801  012443 G      ER9007  017036 G
BIT3    = 000010 G    C$GETB  = 000026      EF9010  004710 G    EM1901  012517 G      ER9008  017126 G
BIT4    = 000020 G    C$GETW  = 000027      EF9016  004777 G    EM2001  012602 G      ER9101  017222 G
BIT5    = 000040 G    C$GMAN  = 000043      EF9017  005074 G    EM2002  012652 G      ER9301  017262 G
BIT6    = 000100 G    C$GPHR  = 000042      EF9018  005150 G    EM2301  012725 G      EVL     = 000004 G
BIT7    = 000200 G    C$GPLO  = 000030      EF9301  005255 G    EM2302  012772 G      EXOERR  002304 G
BIT8    = 000400 G    C$GPRI  = 000040      EF9302  005333 G    EM2401  013030 G      F$END   = 002100
BIT9    = 001000 G    C$INIT  = 000011      EM0101  020360 G    EM2601  013073 G      F$LOAD  = 000035
BMPCQB  002526 G      C$INLP  = 000020      EM0102  020444 G    EM2602  013151 G      FDATA   002250 G
BMPCQE  002726 G      C$MANI  = 000050      EM0103  005560 G    EM2603  013222 G      FDLSA   002250 G
BMPCQP  002524 G      C$MEM   = 000031      EM0201  005616 G    EM2604  013310 G      FDLSO   = 000006 G
BOE     = 000400 G    C$MSG   = 000023      EM0202  005651 G    EM2605  013404 G      F$AU    = 000015
PRLEVL  002240 G      C$OPEN  = 000034      EM0203  006024 G    EM2606  013467 G      F$AUTO  = 000020
BUFBAS  002726 G      C$PNTB  = 000014      EM0204  006167 G    EM2607  013526 G      F$BGN   = 000040
BUFEND  003726 G      C$PNTF  = 000017      EM0301  006346 G    EM2608  013622 G      F$CLEA  = 000012
BUFMID  003526 G      C$PNTS  = 000016      EM0302  006420 G    EM2609  013673 G      F$DU    = 000016
BUFPTR  002302 G      C$PNTX  = 000015      EM0303  006560 G    EM2610  013752 G      F$END   = 000041
BUFSQT  003526 G      C$QIO   = 000377      EM0401  006757 G    EM2611  014044 G      F$HARD  = 000004
CACHRX  023632 G      C$RDBU  = 000007      EM0402  007015 G    EM2612  014127 G      F$HW    = 000015
CACHTX  023660 G      C$REFG  = 000047      EM0501  007165 G    EM3001  014223 G      F$INIT  = 000006
CALMSL  017526 G      C$RESE  = 000033      EM0502  007241 G    EM3002  014262 G      F$JMP   = 000050
CKTRAP  017752 G      C$REVI  = 000003      EM0505  007415 G    EM3003  014356 G      F$MOD   = 000000
CLKBRL  002336 G      C$RFLA  = 000021      EM0506  007505 G    EM3004  014442 G      F$MSG   = 000011
CLKCSR  002334 G      C$RPT   = 000025      EM0601  007575 G    EM3005  014506 G      F$PROT  = 000001
CLKHRZ  002342 G      C$SEFG  = 000046      EM0602  007657 G    EM3101  014600 G      F$PWR   = 000017
CLKINT  023706 G      C$SPRI  = 000041      EM0603  010017 G    EM3102  014645 G      F$RPT   = 000012
CLKVEC  002340 G      C$SVEC  = 000037      EM0701  010202 G    EM9014  014722 G      F$SEG   = 000005
CLNRST  020002 G      C$TPRI  = 000015      EM0702  010245 G    EM9017  015016 G      F$SOFT  = 000009
CLR16W  020024 G      DELAY   020124 G      EM0703  010425 G    EM9018  015122 G      F$SRV   = 000010
CNTERR  020046 G      DPTBL   002214 G      EM0801  010610 G    EM9019  015157 G      F$SUB   = 000007
CSRA    002242 G      DIAGMC  = 000000      EM0802  010656 G    EM9020  015154 G      F$SW    = 000014
CTRLCF  002306 G      DRADRT  002242 G      EM0901  010726 G    EM9022  015200 G      F$TEST  = 000001
```

SYMBOL TABLE

```
GETPRM  024640          LPRO   = 000004 G      L10002  015624          MSLOOP  020300 G      SPPTBL  002224 G
GPRS08  002450 G        L$ACP   002110 G      L10003  016134          MSTICK  002354 G      SKPSTS  022344 G
G$CNTO= 000200          L$APT   002036 G      L10004  016172          NDERPT  002226 G      SVCGBL= 000000
G$DELM= 000372          L$AU    025206 G      L10005  016252          NEWPAS  024620         SVCINS= 000001
G$DISP= 000003          L$AUT   002070 G      L10006  016414          NEWRES  024612         SVCSUB= 000001
G$EXCP= 000400          L$AUTO  025056 G      L10007  016524          NEWSTA  024302         SVCTAG= 000001
G$HILI= 000002          L$CCP   002106 G      L10010  016616          NUMLNS= 000020 G       SVCTST= 000001
G$LOLI= 000001          L$CLEA  025060 G      L10011  016720          OOPS    020514 G       SWAPO   022422 G
G$NO   = 000000         L$CO    002032 G      L10012  017034          OPTION  002224 G       SWPTQ1  037130
G$OFFS= 000400          L$DEPO  002011 G      L10013  017124          O$APTS= 000000         SWPTQ2  037204
G$OFST= 000376          L$DESC  004046 G      L10014  017220          O$AU   = 000000         SWPTQ3  037255
G$PRMA= 000001          L$DESP  002076 G      L10015  017260          O$BGNR= 000001         SWPTQ4  037310
G$PRMD= 000002          L$DEVP  002060 G      L10016  017452          O$BGNS= 000001         SYSGEN= 010000
G$PRM  = 000000         L$DISP  002124 G      L10017  024226          O$DU   = 000001         TIMER1  002344 G
G$RADA= 000140          L$DLY   002116 G      L10021  025054          O$ERRT= 000001         TIMER2  002346 G
G$RADB= 000000          L$DTP   002040 G      L10022  025072          O$GNSW= 000001         TIMER3  002350 G
G$RADD= 000040          L$DTYP  002034 G      L10023  025074          O$POIN= 000001         TNUM   = 000032 G
G$RADL= 000120          L$DU    025076 G      L10024  025204          O$SETU= 000000         TP4FLG  002320 G
G$RADO= 000020          L$DUT   002072 G      L10025  025212          PARDA   002366 G       TP4RTN  024124 G
G$XFER= 000004          L$DVTY  004036 G      L10026  025474          PASCNT  002312 G       TP4VEC  002322 G
G$YES  = 000010         L$EF    002052 G      L10027  025724          PCSLOT= 000016 G       TSABRT  022472 G
HELP   = 000000         L$ENVI  002044 G      L10030  026170          PNT    = 001000 G      TSTNUM  002324 G
HOE    = 100000 G       L$ERRT  003766 G      L10031  026366          PREGRT  004020 G       TXAD1A  002254 G
HWPTQ1  036744          L$ETP   002102 G      L10032  026560          PREG05  003776         TXAD2A  002256 G
HWPTQ2  036762          L$EXP1  002046 G      L10033  025776          PRI    = 002000 G      TXBFCA  002260 G
HWPTQ3  037015          L$EXP4  002064 G      L10034  027204          PRI00  = 000000 G      TXBFCO= 000016 G
HWPTQ4  037043          L$EXP5  002066 G      L10035  027412          PRI01  = 000040 G      TXDSBL  022604 G
IBF    = 010000 G       L$HARD  036700 G      L10036  027606          PRI02  = 000100 G      TXENBL  022700 G
IDU    = 000040 G       L$HIME  002120 G      L10037  030022          PRI03  = 000140 G      TXIE0   022774 G
IER    = 020000 G       L$HPCP  002016 G      L10040  030250          PRI04  = 000200 G      TXIE1   023054 G
IESTAT  002510 G        L$HPTP  002022 G      L10041  030544          PRI05  = 000240 G      TXINTC  002326 G
ISR    = 000100 G       L$HW    002212 G      L10042  031032          PRI06  = 000300 G      TXINTF  002330 G
IXE    = 004000 G       L$ICP   002104 G      L10043  031432          PRI07  = 000340 G      TXINTR  024146 G
I$AU   = 000041         L$INIT  024236 G      L10044  031774          PUFIFO  020542 G       TXVECA  002234 G
I$AUTO= 000041          L$LADP  002026 G      L10045  032242          RBUFA   002244 G       T$ARGC= 000003
I$CLN  = 000041         L$LAST  037454 G      L10046  032432          RDPDR   020624 G       T$CODE= 001052
I$DU   = 000041         L$LOAD  002100 G      L10047  032750          REGTST  021076 G       T$ERRN= 022105
I$HRD  = 000041         L$LUN   002074 G      L10050  033212          REPSMR  021330 G       T$EXCP= 000000
I$INIT= 000041          L$MREV  002050 G      L10051  033326          RESETT  021356 G       T$FLAG= 000050
I$MOD  = 000041         L$NAME  002000 G      L10052  033644          RMATBB  002262 G       T$GMAN= 000000
I$MSG  = 000041         L$PRIO  002042 G      L10053  034220          ROLDAP  021470 G       T$HILI= 177777
I$PROT= 000040          L$PROT  024230 G      L10054  035556          RSTRPT  021514 G       T$LAST= 000001
I$PTAB= 000041          L$PRT   002112 G      L10055  036320          RXBRRT  023756 G       T$LOLI= 000000
I$PWR  = 000041         L$REPP  002062 G      L10056  036614          RXIE0   022212 G       T$SYSM= 010000
I$RPT  = 000041         L$REV   002010 G      L10057  036674          RXIE1   022252 G       T$UTNO= 000032
I$SEG  = 000041         L$RPT   024222 G      L10060  036744          RXINPT  024036 G       T$NEST= 177777
I$SETU= 000041          L$SOFT  037072 G      L10061  037130          RXINTC  002314 G       T$NSO  = 000000
I$SPT  = 000041         L$SPC   002056 G      MAPLNS= 177777 G        RXINTF  002316 G       T$NS1  = 000005
I$SRV  = 000041         L$SPCP  002020 G      MPUNIT  005455 G        RXTMA   002244 G       T$PTNO= 000000
I$SUB  = 000041         L$SPTP  002024 G      MMENAB  002364 G        RXVECA  002252 G       T$SAVL= 177777
I$TST  = 000041         L$STA   002030 G      MMPRES  002362 G        ROSLOT= 000002 G       T$SEGL= 177777
JSJMP  = 000167         L$SW    002224 G      MMSR0   002360 G        R1SLOT= 000004 G       T$SUBN= 000000
LNCTRA  002252 G        L$TEST  002114 G      MSG1    015626 G        R2SLOT= 000006 G       T$TAGL= 177777
LOE    = 040000 G       L$TIML  002014 G      MSG2    015704 G        R3SLOT= 000010 G       T$TAGN= 010063
LOT    = 000010 G       L$UNIT  002012 G      MSG5    015765 G        R4SLOT= 000012 G       T$TEMP= 000000
LPCSLT= 000036 G        L10000  002220         MSLCNT  002356 G        R5SLOT= 000014 G       T$TEST= 000032
LPRA    002246 G        L10001  002230         MSLGET  020164 G        SAVDMP  022276 G       T$TSTM= 177777
```

SYMBOL TABLE

```
T$TSTS= 000001        T$$SOF= 010061        T17     032244 G      T3      025726 G      WAIBIC  023214 G
T$$AUD= 010025        T$$SW = 010001        T18     032434 G      T4      026172 G      WAIBIS  023270 G
T$$AUT= 010022        T$$TES= 010057        T19     032752 G      T5      026370 G      WDPDR   023344 G
T$$CLE= 010023        T1      025214 G      T2      025476 G      T6      026562 G      WORD1   002332 G
T$$DU = 010024        T10     027610 G      T20     033214 G      T7      027000 G      WTWLNC  023552 G
T$$HAR= 010060        T11     030024 G      T21     033330 G      T8      027206 G      WTWLPR  023602 G
T$$HW = 010000        T12     030252 G      T22     033646 G      T9      027414 G      X$ALWA= 000000
T$$INI= 010021        T13     030546 G      T23     034222 G      UAM     000200 G      X$FALS= 000040
T$$MSG= 010016        T14     031034 G      T24     035360 G      UNBTTB  002370 G      X$OFFS= 000400
T$$PRO= 010020        T15     031434 G      T25     036322 G      UNITN   002236 G      X$TRUE= 000020
T$$RPT= 010017        T16     031776 G      T26     036616 G      UNSDIV  023060 G      $PATCH  037400 G

. ABS.  053454      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  28661 WORDS  ( 112 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:05:14
C2DHUB0,BIN,C2DHUB0,LST/-SP=SVC34R/ML,C2DHUB0,P11
```