

DHU11

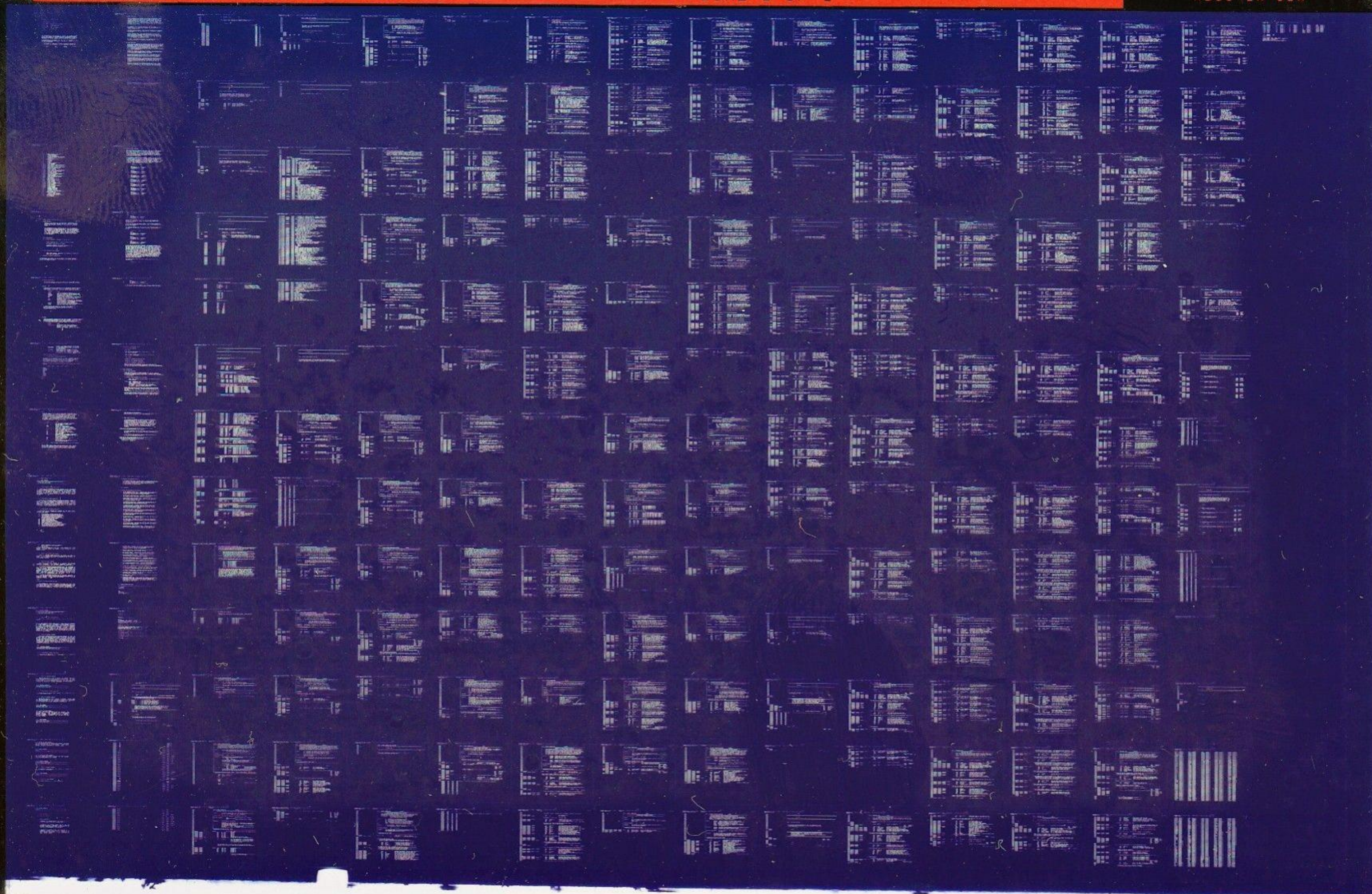
DHU-11 FUNC TST PART 1
CZDHUAO

COPYRIGHT (c) 1983-84
AH-T795A-MC
FICHE 1 OF 1

APR 1984

digital

Made In USA



.REM 6

IDENTIFICATION

PRODUCT CODE: AC-T794A-MC
PRODUCT NAME: CZDHUAO DHU-11 FUNC TST PART1
PRODUCT DATE: 15 DECEMBER 1983
MAINTAINER: ENE DIAGNOSTICS GROUP
AUTHOR: ANTHONY HART
MODIFIED BY:

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1983,1984 BY DIGITAL EQUIPMENT CORPORATION
THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

C1

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

ORIGINAL RELEASE: 15 DEC 83 ANTHONY HART

TABLE OF CONTENTS

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

1.0 GENERAL PROGRAM CONSIDERATIONS

1.1 PROGRAM ABSTRACT

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

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

1.2 SYSTEM REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DHU FVT:

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

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCHY 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 PERFORMANCE AND PROGRESS REPORTS SECTION OF THIS DOCUMENT)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE FLAGS SECTION)
ZFLAGS	CLEAR ALL FLAGS (SEE FLAGS SECTION)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO
YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".
MORE INFORMATION CAN BE FOUND WITHIN THE SECTION LABELLED
EXTENDED COMMAND SYNTAX

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION.
THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL
SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH.
IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDDD".

SWITCH	EFFECT
-----	-----
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO

BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
 EXECUTE DDDDD PASSES (DDDD = 1 TO 64000)
 SET SPECIFIED FLAGS. SEE THE FLAGS SECTION
 OF THIS DOCUMENT.
 REPORT END OF PASS MESSAGE AFTER EVERY
 DDDDD PASSES ONLY. (DDDD = 1 TO 64000)
 TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED
 IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12
 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

START/TESTS:1-5/PASS:1000/EOP:100

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE
 EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF
 PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A
 SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY,
 FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH
 COMMAND.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
---	-----
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXR*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST
EVL	EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)

*SEE THE ERROR INFORMATION SECTION OF THIS DOCUMENT.

SEE THE XXDP* USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A "BELL" ON ERROR, YOU MAY USE THE FOLLOWING STRING:

/FLAGS:LOE:IER:BOE

2.4 EXTENDED COMMAND SYNTAX

2.4.1 START COMMAND -

```
*****  
STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:  
  <FLAG-LIST>/EOP:<INCR>  
*****
```

2.4.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>) -

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.), SEPERATED BY COLONS, THAT SPECIFY THE TESTS TO BE EXECUTED. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF "EFFECT OF START COMMAND" SECTION.

2.4.1.2 PASS SWITCH (/PASS:<PASS-CNT>) -

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS). THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE, EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF "EFFECT OF START COMMAND" SECTION.

2.4.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>) -

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPERATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED.
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR.
IER	INHIBIT ERROR REPORTING.
IBE	INHIBIT BASIC ERROR REPORTS.
IXE	INHIBIT EXTENDED ERROR REPORTS.
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER.
PNT	PRINT NUMBER OF TEST BEING EXECUTED.
BOE	BELL ON ERROR (NOT RELATED TO BELL PROMPTING).
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION (ILLEGAL FOR THIS DIAGNOSTIC).
ISR	INHIBIT STATISTICAL REPORTS.

IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC.
(HAS NO EFFECT IN THIS DIAGNOSTIC.)
LOT LOOP ON TEST.
THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE
CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT
GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF "EFFECT OF START
COMMAND" SECTION.

2.4.1.4 END OF PASS SWITCH (/EOP:<INCR>) -

<INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF
PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE
DEFAULT IS AT THE END OF EVERY PASS. SEE EXAMPLE AT END OF "EFFECT OF
START COMMAND" SECTION.

2.4.1.5 EFFECT OF START COMMAND -

THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE
PARAMETER DIALOGUE, THE SOFTWARE PARAMETER DIALOGUE, THE
INITIALIZATION QUESTIONS, AND THEN THE DIAGNOSTIC COMMENCES TESTING.

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION "0
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:MOE=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 NUMP INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIALOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

2.4.2.3 EFFECT OF RESTART COMMAND -

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT THE P TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH SHOULD NOT BE USED WITH THIS PROGRAM. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE, B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET, OR C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

2.4.3 CONTINUE COMMAND -

```
*****  
CON(TINUE)/PASS:<PASS-CNT>/FLAGS:<FLAG-LIST>  
*****
```

2.4.3.1 FLAG SWITCH (/FLAGS:<FLAG-LIST>) -

<FLAG-LIST> IS SAME AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

2.4.3.2 EFFECT OF CONTINUE COMMAND -

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

2.4.4 PROCEED COMMAND -

PRO(CEED)/FLAGS:<FLAG-LIST>

2.4.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>) -

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

2.4.4.2 EFFECT OF PROCEED COMMAND -

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

2.4.5 ADD COMMAND -

ADD/UNITS:<UNIT-LIST>

2.4.6 EFFECT OF ADD COMMAND -

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

2.4.7 DROP COMMAND -

DRO(P)/UNITS:<UNIT-LIST>

2.4.8 EFFECT OF DROP COMMAND
THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS
WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START
COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND
MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

2.4.9 PRINT COMMAND -

PRI(NT)

2.4.9.1 EFFECT OF PRINT COMMAND -
THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST
START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT
STATISTICAL REPORTING) FLAG IS CLEARED.

2.4.10 DISPLAY COMMAND -

DIS(PLAY)/UNITS:<UNIT-LIST>

2.4.10.1 EFFECT OF DISPLAY COMMAND -
THE HARDWARE P-TABLE FOR THE TEST STATION IS PRINTED IN THE
FORMAT IN WHICH IT WAS ENTERED.

2.4.11 FLAGS COMMAND -

FLA(GS)

2.4.11.1 EFFECT OF FLAGS COMMAND -
THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

2.4.12 ZFLAGS COMMAND

ZFL(AGS)

2.4.13 ZFLAGS COMMAND -

ALL FLAGS ARE CLEARED.

2.4.14 CONTROL CHARACTERS -

- C A CONTROL/C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.
- Z A CONTROL/Z (Z) ENTERED DURING ONE OF THE TWO OPERATOR DIALOGUES-- HARDWARE P-TABLE DIALOGUE OR SOFTWARE P-TABLE DIALOGUE CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.
- O A CONTROL/O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES ALL TELETYPE OUTPUT TO BE SUPPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER CONTROL/O IS TYPED, WHICH RESTORES NORMAL TELETYPE OUTPUT.

2.5 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?" YOU MUST ANSWER "Y" AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 6 OF THE XXDP+ USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A "Y", THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL). YOU WILL THEN BE ASKED THE FOLLOWING QUESTIONS FOR EACH UNIT.

1. CSR ADDRESS - THIS QUESTION REQUESTS THE CSR ADDRESS OF THE SPECIFIED DMU-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 DMU-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 DMU11 WHICH ARE BEING SELECTED FOR TESTING. IF THE BIT IN THE BIT MAP IS SET WHICH CORRESPONDS TO A PARTICULAR LINE (I.E. BIT 5 FOR LINE 5) THAT LINE WILL BE TESTED BY THE FVT.
4. BR LEVEL - THIS QUESTION REQUESTS THE INTERRUPT BR LEVEL OF THE SPECIFIED DMU-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

CZDHLA0 DHU 11 FUNC TST PART1 MACRO M1200 12 DEC 83 16:08 PAGE 15 1
PROGRAM DOCUMENT

SEO 15

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 (0) ? 8<CR>

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

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

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

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

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

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

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

```
UNIT 8
CSR ADDRESS (0) 160000<CR>
SUB-DEVICE # (0) ? 7<CR>
Q-FACTOR (0) 1 ? <CR>
```

NOTICE THAT THE DEFAULT VALUE FOR THE Q-FACTOR CHANGES WHEN A NON-DEFAULT RESPONSE IS GIVEN. BE CAREFUL WHEN SPECIFYING MULTIPLE UNITS!

AS YOU CAN SEE FROM THE ABOVE EXAMPLE, THE HARDWARE PARAMETERS DO NOT VARY SIGNIFICANTLY FROM UNIT TO UNIT. THE PROCEDURE SHOWN IS NOT VERY EFFICIENT.

THE RUNTIME SERVICES CAN TAKE MULTIPLE UNIT SPECIFICATIONS HOWEVER. LET'S BUILD THE SAME TABLE USING THE MULTIPLE SPECIFICATION FEATURE.

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

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

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

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

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

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

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

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

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

2.8 QUICK START UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

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

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

THE GENERAL ERROR MESSAGE IS OF THE FORM:

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

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

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

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

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

3.2 SPECIFIC ERROR MESSAGES

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

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

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

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

CZDHU DVC FTL ERR 01603 ON UNIT 02 TST 15 SUB 000 PC: XXXXXX
DEVICE REGISTER WORD READ/WRITE TEST FAILED.
BAD BIT(S) IN DEVICE TBUFAD1 REGISTER FOR LINE 7 (D).
EXPECTED DATA: 000000 (0).
ACTUAL DATA: 000023 (0).

4.0 PERFORMANCE AND PROGRESS REPORTS

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

5.0 TEST SUMMARIES

THE FOLLOWING ARE INCLUDED WITHIN CZDMUA:

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

12. SELFTEST TEST - VERIFIES THAT THE DUT'S SELFTEST EXECUTES WITHIN THE CORRECT TIME AND THAT THE CORRECT CODES ARE RETURNED IN THE FIFO AFTER ITS COMPLEATION.
13. SELFTEST FAIL TEST - VERIFIES THAT THE DUT WILL REPORT ERRORS CORRECTLY WHEN IT IS FORCED TO FAIL.
14. ROM VERSION NUMBER - VERIFIES THAT THE ROM VERSION NUMBERS ARE REPORTED CORRECTLY AND IF REQUESTED PRINTS THEM OUT.
15. WORD ACCESS READ/WRITE TEST - VERIFIES THAT THE REGISTERS RESPOND CORRECTLY TO READ AND WRITE ACCESSES.
16. WORD ACCESS READ/MODIFY/WRITE TEST - VERIFIES THAT THE REGISTERS WILL RESPOND CORRECTLY TO READ/MODIFY/WRITE ACCESES.
17. BYTE ACCESS READ/WRITE TEST - VERIFIFS THAT THE REGISTERS WILL RESPOND CORRECTLY TO BYTE READ/WRITE ACCESSES.
18. BYTE ACCESS READ/MODIFY/WRITE - VERIFIES THAT THE REGISTERS WILL RESPOND CORRECTLY TO BYTE READ/MODIFY/WRITE ACCESSES.
19. ID.BIT TEST - VERIFIES THAT THE ID BIT READS AS SET.
20. TX.ENABLE (INACTIVE) TEST - VERIFIES THAT WHEN A LINE'S TX.ENBL BIT IS CLEAR, TRANSMISSION WILL NOT TAKE PLACE ON THAT LINE.
21. TX.ENABLE (ACTIVE) TEST - VERIFIES THAT WHEN A LINE'S TX.ENBL BIT IS SET, TRANSMISSION WILL TAKE PLACE ON THAT LINE.
22. INTERRUPT TEST - VERIFIES THAT THE DUT WILL GENERATE RECEPTION AND TRANSMISSION INTERRUPTS CORRECTLY.
23. BR LEVEL TEST - VERIFIES THAT THE DUT INTERRUPTS AT THE CORRECT BUS REQUEST LEVEL.
24. REPORT BMP CODES TEST - THIS PSEUDO TEST REPORTS THE FIRST 32 CHARACTERS WHICH WERE DISCOVERED IN THE FIFO DURING THE EXECUTION OF THE OTHER TESTS. THIS AVOIDS INTERRUPTION OF THE OTHER TESTS BY THESE CODES IF THEY ARE NOT CRITICAL TO THE PERFORMANCE OF THE TESTS.

6.0 EXAMPLE ERROR FREE PASS

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

.R CZDHUAO
CZDHUAO.BIN

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

RESTRT ADDR: 147670
DR>STA/PAS:1

CHANGE HW (L) ? Y

* UNITS (D) ? 2

UNIT 0
CSR ADDRESS: (0) 160460 ? +Z

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

CHANGE SW (L) ? Y

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

TESTING UNIT : 0(D):

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

TESTING UNIT : 1(D)

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

CZDHU EOP 1
 0 TOTAL ERRS

DR>

E

```
1048      .LIST SEQ,LOC,BIN,MEB
1049      .NLIST CND
1050
1051      ;*****
1052      ;
1053      ;          FVTA.PHD
1054      ;
1055      ;*****
1056
1057
1058
1059      .SBTTL  PROGRAM HEADER
1060
1061
1062      .MCALL  SVC
1063 000000      SVC          ; INITIALIZE SUPERVISOR MACROS
1064
1065      ;*****
1066      ;          IF STRUCTURED MACROS ARE TO BE USED, ADD ".MCALL STRUCT" AND "STRUCT"
1067      ;          TO INITIALIZE THE STRUCTURED MACROS.
1068
1069      000001      SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
1070      000001      SVCTST= 1     ; LIST TEST TAGS, SHIFTED RIGHT
1071      000001      SVCSUB= 1     ; LIST SUBTEST TAGS, SHIFTED RIGHT
1072      000001      SVCGBL= 1    ; LIST GLOBAL TAGS, SHIFTED RIGHT
1073      000001      SVCTAG= 1    ; LIST OTHER TAGS, SHIFTED RIGHT
1074
1075      ;          CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1076      ;          TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
1077      ;          SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
1078      ;          CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1079      ;*****
1080
1081 000000      .ENABL  ABS
1082      .ENABL  AMA
1083      "      "      2000
1084
1085 002000      BGNMOD
1086
1087      ;**
1088      ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1089      ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1090      ;--
1091
1092 002000      POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL
1093
1110
1111 002000      HEADER  CZDHU,A,0,16,0,PRI07
1112      002000
1113      002000      103
1114      002001      132
```

```
L0NAME::
        .ASCII /C/
        .ASCII /Z/
```

```

002002      104
002003      110
002004      125
002005      000
002006      000
002007      000
002010
002010      101
002011
002011      060
002012
002012      000000
002014
002014      000016
002016
002016      035722
002020
002020      036114
002022
002022      002206
002024
002024      002220
002026
002026      036476
002030
002030      000000
002032
002032      000000
002034
002034      000000
002036
002036      000000
002040
002040      002124
002042
002042      000340
002044
002044      000000
002046
002046      000000
002050
002050      003
002051      003
002052
002052      000000
002054
002054      000000
002056
002056      000000
002060
002060      004032
002062
002062      024102
002064
002064      000000
002066
002066      000000
002070

```

```

      .ASCII /D/
      .ASCII /H/
      .ASCII /U/
      .BYTE 0
      .BYTE 0
      .BYTE 0
L$REV::
      .ASCII /A/
L$DEPO::
      .ASCII /O/
L$UNIT::
      .WORD 0
L$TIML::
      .WORD 16
L$HPCP::
      .WORD L$HARD
L$SPCP::
      .WORD L$SOFT
L$HPTP::
      .WORD L$HW
L$SPTP::
      .WORD L$SW
L$LADP::
      .WORD L$LAST
L$STA::
      .WORD 0
L$CO::
      .WORD 0
L$DTYP::
      .WORD 0
L$APT::
      .WORD 0
L$DTP::
      .WORD L$DISPATCH
L$PRIO::
      .WORD PRI07
L$ENVI::
      .WORD 0
L$EXP1::
      .WORD 0
L$MREV::
      .BYTE C$REVISION
      .BYTE C$EDIT
L$Fr::
      .WORD 0
      .WORD 0
L$SPC::
      .WORD 0
L$DEVP::
      .WORD L$DVTYP
L$REPP::
      .WORD L$RPT
L$EXP4::
      .WORD 0
L$EXP5::
      .WORD 0
L$AUT::

```

PROGRAM HEADER

002070	000000
002072	
002072	024756
002074	
002074	000000
002076	
002076	004042
002100	
002100	104035
002102	
002102	003762
002104	
002104	024116
002106	
002106	024740
002110	
002110	024736
002112	
002112	024110
002114	
002114	000000
002116	
002116	000000
002120	
002120	000000

1112

L\$DUT::	.WORD	0
L\$DU	.WORD	L\$DU
L\$LUN::	.WORD	0
L\$DESP::	.WORD	L\$DESC
L\$LOAD::	EMT	E\$LOAD
L\$ETP::	.WORD	L\$ERRTBL
L\$ICP::	.WORD	L\$INIT
L\$CCP::	.WORD	L\$CLEAN
L\$ACP::	.WORD	L\$AUTO
L\$PRT::	.WORD	L\$PROT
L\$TEST::	.WORD	0
L\$DLY::	.WORD	0
L\$HIME::	.WORD	0

1124
1125
1126
1127
1128
1129
1130
1131

.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 24

```

002122 000030
002124 025074
002126 025356
002130 025606
002132 026052
002134 026250
002136 026442
002140 026660
002142 027066
002144 027274
002146 027470
002150 027704
002152 030132
002154 030426
002156 030714
002160 031314
002162 031562
002164 031752
002166 032270
002170 032532
002172 032646
002174 033164
002176 033540
002200 034676
002202 035640

```

1132

```

          .WORD 24
L$DISPATCH::
          .WORD T1
          .WORD T2
          .WORD T3
          .WORD T4
          .WORD T5
          .WORD T6
          .WORD T7
          .WORD T8
          .WORD T9
          .WORD T10
          .WORD T11
          .WORD T12
          .WORD T13
          .WORD T14
          .WORD T15
          .WORD T16
          .WORD T17
          .WORD T18
          .WORD T19
          .WORD T20
          .WORD T21
          .WORD T22
          .WORD T23
          .WORD T24

```

1140 ;*****
1141 ;
1142 ; FVTA.DHT
1143 ;
1144 ;*****

1145
1146
1147
1148 .SBTTL DEFAULT HARDWARE P-TABLE
1149
1150 ;**
1151 ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1152 ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
1153 ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P TABLES,
1154 ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P TABLES.
1155 ;**
1156
1157 BGNHW DFPTBL
002204
002204 000004
002206
002206
1158
1159 002206 160460 .WORD 160460 ;DEFAULT CSR ADDRESS
1160 002210 000310 .WORD 310 ;DEFAULT VECTOR ADDRESS
1161 002212 177777 .WORD 177777 ;DEFAULT ACTIVE LINES BIT MAP
1162 002214 005 .BYTE 5 ;DEFAULT BR LEVEL
1163 .EVEN
1164
1165 002216
002216
ENDHW
L10000:
L10000-L10000/2
L10000:
DFPTBL::
L10000-L10000/2

```
1167
1168 ;*****
1169 ;
1170 ;           FVTA.SWT
1171 ;
1172 ;*****
1173
1174
1175
1176 .SBTTL  SOFTWARE P TABLE
1177
1178 ;**
1179 ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
1180 ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
1181 ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
1182 ; AT RUN TIME.
1183 ;--
1184
1185 002216      BGNSW  SFPTBL
1186 002216      000002
1187 002220
1188 002220
1189
1190 002224
1191 002224
```

.WORD L10001-L\$SW/2
L\$SW::
SFPTBL::

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

ENDSW

L10001:

1192
11931194
1195
1196
1197
1198
11991200
1201
1211
1212
1213
12141215
1216
1217
12181219 000020
1220 1777771221
1222
1223 000004
1224 000006
1225 000016
1226

1241 002224

;*****

;
; FVTA.EQU
;

;*****

.SBTTL GLOBAL EQUATES SECTION

; **
; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
; ARE USED IN MORE THAN ONE TEST.
; --NUMLNS==20 ;NUMBER OF LINES ON DHU11 IS 16.
MAPLNS==177777 ;BIT MAP OF LINES ON DHU11.;***** DEVICE REGISTER OFFSETS FROM THE CSR'S ADDRESS *****
LPRO==4 ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
FSLSO==6 ;FIFOSIZE/STATUS REGISTER OFFSET FROM THE CSR ADDRESS
TXBFCO==16 ;TRANSMIT COUNT REGISTER OFFSET FROM THE CSR ADDRESS

EQUALS

;
; BIT DIFINITIONS

100000	BIT15-- 100000
040000	BIT14-- 40000
020000	BIT13-- 20000
010000	BIT12-- 10000
004000	BIT11-- 4000
002000	BIT10-- 2000
001000	BIT09-- 1000
000400	BIT08-- 400
000200	BIT07-- 200
000100	BIT06-- 100
000040	BIT05-- 40
000020	BIT04-- 20
000010	BIT03-- 10
000004	BIT02-- 4
000002	BIT01-- 2
000001	BIT00-- 1

001000	BIT9-- BIT09
000400	BIT8-- BIT08
000200	BIT7-- BIT07
000100	BIT6-- BIT06
000040	BIT5-- BIT05
000020	BIT4-- BIT04
000010	BIT3-- BIT03
000004	BIT2-- BIT02
000002	BIT1-- BIT01
000001	BIT0-- BIT00

```

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

```

```

1244
1245
1246 ;*****
1247 ;
1248 ;           FVTA.GDT
1249 ;
1250 ;*****
1251
1252
1253 .SBTTL  GLOBAL DATA SECTION
1254
1255 ;**
1256 ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1257 ; IN MORE THAN ONE TEST.
1258 ;--
1259
1260
1261 ;*****
1262 ;           UNIT VARIABLE AREA
1263 ;*****
1264
1265 002224 177777          ACTLNS:: .WORD 177777 ;ACTIVE LINE BIT MAP.
1266 002226 000300          RXVECA:: .WORD 300   ;RX VECTOR ADDRESS.
1267 002230 000304          TXVECA:: .WORD 304   ;TX VECTOR ADDRESS.
1268 002232 000000          UNITN:: .WORD 0      ;UNIT NUMBER.
1269 002234 004           BRLEVL:: .BYTE 4      ;INTERRUPT BUS REQUEST LEVEL
1270                                     .EVEN
1271
1272 ;*****
1273 ;           DEVICE REGISTER ADDRESS TABLE
1274 ;*****
1275 002236
1276 002236 160020          CSRA:: .WORD 160020 ;DHU-11 CSR ADDRESS.
1277 002240 160022          RXTMA:: RBUFA:: .WORD 160022 ;DHU-11 RECIEVE BUFFER/TIMER ADDRESS.
1278 002242 160024          LPRA:: .WORD 160024 ;DHU-11 LINE PARAMETER REGISTER ADDRESS.
1279 002244 160026          FDATA:: FLSA:: .WORD 160026 ;DHU-11 FIFO SIZE/LINE STATUS REGISTER ADDRESS.
1280                                     ;AND FIFO DATA REGISTER ADDRESS.
1281 002246 160030          LNCTRA:: .WORD 160030 ;DHU-11 LINE CONTROL REGISTER ADDRESS.
1282 002250 160032          TXAD1A:: .WORD 160032 ;DHU-11 TRANSMIT BUFFER 1 REGISTER ADDRESS
1283 002252 160034          TXAD2A:: .WORD 160034 ;DHU-11 TRANSMIT BUFFER 2 REGISTER ADDRESS
1284 002254 160036          TXBFCA:: .WORD 16003 ;DHU-11 TRANSMIT BUFFER COUNT REGISTER ADDRESS
1285
1286 ;*****
1287 ;           REGISTER MESSAGE ADDRESS TABLE
1288 ;*****
1289 002256 005470          RMATB8:: .WORD DR00MG ;ADDRESS OF "CSR" MESSAGE.
1290 002260 005474          .WORD DR02MG ;ADDRESS OF "RBUF" MESSAGE.
1291 002262 005501          .WORD DR04MG ;ADDRESS OF "LPR" MESSAGE.
1292 002264 005505          .WORD DR06MG ;ADDRESS OF "STAT" MESSAGE.
1293 002266 005523          .WORD DR10MG ;ADDRESS OF "LNCTRL" MESSAGE.
1294 002270 005532          .WORD DR12MG ;ADDRESS OF "TBUFFAD1" MESSAGE.
1295 002272 005543          .WORD DR14MG ;ADDRESS OF "TBUFFAD2" MESSAGE.
1296 002274 005554          .WORD DR16MG ;ADDRESS OF "TBUFFCT" MESSAGE.
1297
1298 ;*****
1299 ;           ASSORTED GLOBAL VARIABLES:
1300 ;*****

```

```

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

```

```

1358 002420 000100      .WORD 100      ;BIT 6 SET.
1359 002422 000200      .WORD 200      ;BIT 7 SET.
1360 002424 000400      .WORD 400      ;BIT 8 SET.
1361 002426 001000      .WORD 1000     ;BIT 9 SET.
1362 002430 002000      .WORD 2000     ;BIT 10 SET.
1363 002432 004000      .WORD 4000     ;BIT 11 SET.
1364 002434 010000      .WORD 10000    ;BIT 12 SET.
1365 002436 020000      .WORD 20000    ;BIT 13 SET.
1366 002440 040000      .WORD 40000    ;BIT 14 SET.
1367 002442 100000      .WORD 100000   ;BIT 15 SET.
1368
1369
1370
1371
1372 002444
1373 002444 000000      ;*****
1374 002446 000000      ;*      GPR SAVE AREA ZERO.
1375 002450 000000      ;*****
1376 002452 000000      GPRS0B::      ;BASE OF GPR SAVE AREA NUMBER ZERO.
1377 002454 000000      .WORD 0      ;WORD 1, STORAGE FOR R1.
1378
1379
1380
1381
1382 002456 000000      .WORD 0      ;WORD 2, STORAGE FOR R2.
1383 002460
1384
1385
1386
1387
1388 002520 000000      .WORD 0      ;WORD 3, STORAGE FOR R3.
1389 002522
1390 002722
1391
1392
1393
1394
1395 002722
1396 002722
1397 003322
1398 003522
1399 003722
1400 003722
1401
1402
1403
1416 003762
1417
1418

```

TRANSMISSION AND RECEPTION VARIABLES, POINTERS, AND FLAGS.

```

ERSMRF:: .WORD 0      ;ERROR SUMMARY REPORT FLAGS.
ERCNTB:: .BLKW 16.    ;TABLE OF ERROR COUNTERS.

```

STORAGE AREA FOR THE BMP CODE QUEUE.

```

BMPCQP:: .WORD 0      ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
BMPCQB:: .BLKW 64.    ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
BMPCQE::      ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.

```

GENERAL TABLE AND BUFFER AREA--513 WORDS.

```

BUFBAS::      ;BASE OF MEMORY BUFFER.
ERLTBL::      .BLKW 128. ;FIRST HALF OF GENERAL TABLE OR BUFFER.
BUFMID::      .BLKW 64.  ;SECOND HALF OF GENERAL TABLE OR BUFFER.
BUF3QT::      .BLKW 64.  ;LAST QUARTER OF THE BUFFER AREA.
BUFEND::      ;END OF GENERAL PURPOSE MEMORY BUFFER.
ENDETB::      .BLKW 16.  ;BUFFER OVERFLOW SPACE.

```

ERRTBL

```

ERRTYP::      .WORD 0
ERRNBR::      .WORD 0
ERRMSG::      .WORD 0
ERRBLK::      .WORD 0

```

L\$ERRTBL::

.EVEN

```

1420 .SBTTL GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.
1421 ;*****
1422 ;* THERE ARE 4 ROUTINES AND MACRO DEFINITIONS USED FOR THE HANDLING OF
1423 ;* GPR VALUES DURING SUBROUTINE CALLS WITHIN THIS PROGRAM. THE FOUR
1424 ;* ROUTINES/MACRO CALLS HAVE THE FOLLOWING NAMES:
1425 ;*
1426 ;* SAVE - MACRO DEFINITION USED AT THE BEGINNING OF A SUBROUTINE TO
1427 ;* SAVE THE GPR CONTENTS FOR LATER RESTORATION.
1428 ;* PASS - MACRO DEFINITION USED AT THE END OF A SUBROUTINE TO RESTORE
1429 ;* THE PREVIOUSLY SAVED GPR CONTENTS AND TO LEAVE THE CONTENTS
1430 ;* OF THE SPECIFIED GPR(S) INTACT (NOT RESTORED).
1431 ;* PREG05 - SUBROUTINE WHICH IS CALLED FROM THE SAVE AND PASS MACRO
1432 ;* EXPANSIONS WHICH ACTUALLY PERFORMS THE ACTIONS ON THE GPRS.
1433 ;*
1434 ;* DURING A SUBROUTINE WHICH USES THESE GPR SAVE ROUTINES THE VALUES
1435 ;* OF THE GPRS ARE STORED ON THE STACK IN THE FOLLOWING STACK FRAME:
1436 ;*
1437 ;* SP -> RET PC INTO PREG05 ROUTINE.
1438 ;* SP+2 -> GPR R0 CONTENTS.
1439 ;* SP+4 -> GPR R1 CONTENTS.
1440 ;* SP+6 -> GPR R2 CONTENTS.
1441 ;* SP+8 -> GPR R3 CONTENTS.
1442 ;* SP+10 -> GPR R4 CONTENTS.
1443 ;* SP+12 -> GPR R5 CONTENTS.
1444 ;* SP+14 -> RET PC INTO CALLER OF SUB'TNE WHICH CALLED PREG05.
1445 ;*
1446 ;* EACH LEVEL OF SUB'TNE CALLING USES 8 WORDS OF STACK OVERHEAD.
1447 ;* THE SAVE AND PASS MACROS CAN ALSO BE USED IN "STRAIGHT LINE CODE"
1448 ;* TO SAVE AND RESTORE THE GPR VALUES. IN ANY CASE, AFTER THE
1449 ;* ISSUING OF A PASS CALL THE GPRS WILL BE RESTORED TO THE VALUES
1450 ;* THEY HAD PRIOR TO THE LAST SAVE CALL (EXCEPT FOR THE EXCEPTED,
1451 ;* OR PASSED INTACT, GPRS SPECIFIED AS PARAMETERS TO THE PASS CALL)
1452 ;* AND THE SP WILL ALSO BE RESTORED TO ITS CONDITION BEFORE THE LAST
1453 ;* SAVE CALL. THE PROGRAMMER MUST BE SURE THAT THE SP HAS THE SAME
1454 ;* VALUE WHEN THE PASS MACRO IS CALLED AS IT HAD IMMEDIATELY AFTER
1455 ;* THE SAVE MACRO WAS CALLED.
1456 ;*****

```

```
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
```

```

.SBTL GPR FRAME ACCESS EQUATES
;***
;EQUATES THAT ALLOW ACCESS TO THE STACK FRAME. THESE ARE THE
;OFFSETS INTO THE STACK FOR REGISTERS SAVED DURING THE PREGOS
;ROUTINE.
;---
LPCSLT== 36 ;OFFSET FOR LAST RETURN PC.
PCSLT== 16 ;OFFSET FOR RETURN PC.
R5SLOT== 14 ;OFFSET FOR R5.
R4SLOT== 12 ;OFFSET FOR R4.
R3SLOT== 10 ;OFFSET FOR R3.
R2SLOT== 6 ;OFFSET FOR R2.
R1SLOT== 4 ;OFFSET FOR R1.
ROSLOT== 2 ;OFFSET FOR R0.
```

```

000036
000016
000014
000012
000010
000006
000004
000002
```

```

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

```

```
1499 .SBTTL GLOBAL MACRO DEFINITION - PASS -
1500 ;*****
1501 ;* THIS MACRO IS USED IN CONJUNCTION WITH THE SAVE MACRO. IT IS
1502 ;* CALLED AT END OF A SUBROUTINE TO PASS PARAMETERS IN GPRS BACK TO THE
1503 ;* CALLING ROUTINE BY ALTERING THE GPR SAVE AREA ON THE STACK AND THEN
1504 ;* RETURNING TO PREG05 TO RESTORE THE GPRS TO THEIR SAVED VALUES.
1505 ;*
1506 ;* INPUTS: ONLY ALLOWED ARGUMENTS ARE "R0" THRU "R5".
1507 ;* ROSLOT THRU R5SLOT MUST BE EQUATED TO THEIR RESPECTIVE GPR SAVE
1508 ;* SLOT OFFSETS BEFORE CALLING THIS MACRO.
1509 ;*
1510 ;* OUTPUTS: THE GPR VALUES ARE PUT IN THEIR RESPECTIVE SLOTS ON THE STACK.
1511 ;*
1512 ;* CALLING SEQUENCE: PASS R0,R1,...
1513 ;*
1514 ;* COMMENTS: ANY COMBINATION OF GPR ARGUMENTS MAY BE LISTED IN ANY ORDER.
1515 ;* FOR EXAMPLE, THE FOLLOWING ARE LEGAL:
1516 ;* PASS R1
1517 ;* PASS R4,R0,R2
1518 ;* THE GPRS LISTED AS ARGUMENTS WILL BE PASSED INTACT TO THE
1519 ;* CALLING ROUTINE, ALL OTHER GPRS WILL BE RESTORED.
1520 ;* THE SP MUST BE AT ITS ORIGINAL VALUE WHEN PASS IS CALLED.
1521 ;*
1522 ;* THE MACRO CALL
1523 ;* PASS R0,R3
1524 ;* EXPANDS INTO THE FOLLOWING ASSEMBLY CODE:
1525 ;* MOV R0,ROSLOT(SP) ;PUT R0 IN STACK SLOT.
1526 ;* MOV R3,R3SLOT(SP) ;PUT R3 IN STACK SLOT.
1527 ;* JSR PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
1528 ;* IN THIS EXAMPLE GPRS R1, R2, R4, AND R5 WILL BE RESTORED TO
1529 ;* THEIR VALUES CONTAINED IN THE STACK FRAME AND R0 AND R3
1530 ;* WILL BE LEFT AT THEIR VALUES PRIOR TO THIS PASS CALL.
1531 ;*
1532 ;* SUBORDINATE ROUTINES CALLED: (PREGRT - LABEL WITHIN PREG05, VALUE ON STACK.)
1533 ;*****
1534
1535 .MACRO PASS A,B,C,D,E,F
1536 .IRP X,<A,B,C,D,E,F>
1537 .IF NB,X
1538 .LIST
1539 MOV X,X'SLOT(SP) ;PUT X IN STACK SLOT.
1540 .NLIST
1541 .ENOC
1542 .ENDM
1543 .LIST
1544 JSR PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
1545 .NLIST
1546 .ENDM PASS
```

```
1548 .SBTTL GLOBAL SUBROUTINE - PREG05 -
1549 ;*****
1550 ;* PRESERVE REGISTERS R0 THROUGH R5 FOR SUBROUTINE CALLS.
1551 ;*
1552 ;* INPUTS: THE RETURN ADDRESS BACK INTO THE CALLING ROUTINE MUST BE IN
1553 ;* GPR R5. (I.E. - MACROS USE "JSR R5,PREG05".)
1554 ;*
1555 ;* OUTPUTS: REGISTERS R0 THROUGH R5 ARE SAVED ON THE STACK.
1556 ;*
1557 ;* CALLING SEQUENCE: SAVE ;MACRO EXPANSION CALLS PREG05.
1558 ;* [SUBROUTINE CODE]...
1559 ;* PASS ;MACRO EXPANSION RECALLS PREG05.
1560 ;*
1561 ;* COMMENTS: THIS ROUTINE IS RE-ENTRANT.
1562 ;*
1563 ;* PARAMETERS MAY BE PASSED OUT OF A SUBROUTINE BY MODIFYING THE
1564 ;* REGISTER SAVE AREA ON THE STACK. USE THE PASS GPRN MACRO
1565 ;* TO RETURN GPR VALUES INTACT.
1566 ;* USE THE RNSLOT OFFSETS FROM THE SP TO PASS OTHER PARAMETERS.
1567 ;* [EXAMPLE: MOV VALUE,RNSLOT(SP) ]
1568 ;* MAKE SURE THE SP IS AT ITS ORIGINAL VALUE WHEN YOU DO THIS.
1569 ;*
1570 ;* SUBORDINATE ROUTINES CALLED: NONE.
1571 ;*****
1572
1573 003772 PREG05: ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
1574 003772 010446 MOV R4,-(SP) ;SAVE R4
1575 003774 010346 MOV R3,-(SP) ;SAVE R3
1576 003776 010246 MOV R2,-(SP) ;SAVE R2
1577 004000 010146 MOV R1,-(SP) ;SAVE R1
1578 004002 010046 MOV R0,-(SP) ;SAVE R0
1579 004004 010546 MOV R5,-(SP) ;PUSH RETURN PC ON TOP OF STACK
1580 004006 016605 000014 MOV R5SLOT(SP),R5 ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS
1581
1582 004012 004736 JSR PC,R(SP)+ ;CALL THE SUBROUTINE AT THE RETURN ADDRESS
1583 ;FROM THE PREG05 CALL, PUTTING THE PRESENT
1584 ;PC ON THE STACK AS A RETURN ADDRESS INTO
1585 ;THIS (PREG05) ROUTINE.
1586
1587 ;***
1588 ;THE FOLLOWING CODE IS EXECUTED WHEN THE CALLING ROUTINE DOES A
1589 ;"RETURN" [JSR PC,R(SP)+] USING THE PC DEPOSITED ON THE STACK ABOVE.
1590 ;---
1591
1592 004014 012605 PREGRT: MOV (SP)+,R5 ;PUT RETURN PC IN R5.
1593 004016 012600 MOV (SP)+,R0 ;RESTORE R0.
1594 004020 012601 MOV (SP)+,R1 ;RESTORE R1.
1595 004022 012602 MOV (SP)+,R2 ;RESTORE R2.
1596 004024 012603 MOV (SP)+,R3 ;RESTORE R3.
1597 004026 012604 MOV (SP)+,R4 ;RESTORE R4.
1598
1599 004030 000205 RTS R5 ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05.
1600 ;RESTORING R5 IN THE PROCESS.
```

```

1602      .SBTTL GLOBAL TEXT SECTION
1604      ;*****
1605      ;
1606      ;          FVTSKL1.P11
1607      ;
1608      ;*****
1610
1611
1612      ;*
1613      ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1614      ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1615      ; MORE THAN ONE TEST.
1616      ;
1617
1618      ;
1619      ; NAMES OF DEVICES SUPPORTED BY PROGRAM
1620      ;
1621      ;          DEVTYP <DMU-11>
1622      ;
1623      ;          L$DVTYP::
1624      ;              .ASCIZ /DMU 11/
1625      ;
1626      ;          .EVEN
1627
1628
1629      ; TEST DESCRIPTION
1630      ;
1631      ;          DESCRIPT      <DMU 11 FUNC TST PART1>
1632      ;
1633      ;          L$DESC::
1634      ;              .ASCIZ /DMU-11 FUNC TST PAR
1635      ;
1636      ;          .EVEN
1637
1638      ;
1639      ;
1640

```

004032				
004032	104	110	125	
004035	055	061	061	
004040	000			
004042				
004042	104	110	125	
004045	055	061	061	
004050	040	106	125	
004053	116	103	040	
004056	124	123	124	
004061	040	120	101	
004064	122	124	061	
004067	000			

T1/

```
1642  
1643 ;*****  
1644 ;  
1645 ;          FVTA.FMT  
1646 ;  
1647 ;*****  
1648  
1649  
1650  
1651 ;  
1652 ; FORMAT STATEMENTS USED IN PRINT CALLS  
1653 ;  
1654  
1655  
1656
```

```

1675 ;*****
1676 ;
1677 ; FVTA.MSG
1678 ;
1679 ;*****
1680
1681
1682 .NLIST BIN
1683 .SBTTL GLOBAL MESSAGE AREA
1684 ; ***** FORMAT STATEMENTS *****
1685 004070 MFUNIT:: .ASCIZ /%A TESTING UNIT :%D4%N/
1686 004121 EF0503:: .ASCIZ /%T%N/
1687 004126 EF0505:: .ASCIZ /%A %D5%A ILLEGAL INTERRUPTS RECEIVED.%N/
1688 004201 EF1401:: .ASCIZ /%A ROM VERSION NUMBERS; PROC_1 = %D2%A(D) PROC_2 = %D2%A(D)%N/
1689 004303 EF1402:: .ASCIZ /%T%A ROM VERSION NUMBER %T%N/
1690 004340 EF1601:: .ASCIZ /%A %T%A, TEST ABORTED %N/
1691 004372 EF1602:: .ASCIZ /%A EXPECTED DATA: %D6%A (0).%N/
1692 004434 EF1603:: .ASCIZ /%A ACTUAL DATA: %D6%A (0).%N/
1693 004476 EF1604:: .ASCIZ /%A BAD BIT(S) IN DEVICE %T%A REGISTER FOR LINE %D2%A (D).%N/
1694 004573 EF3001:: .ASCIZ /%A EXPECTED OR CORRECT VALUE: %D3%N/
1695 004642 EF3002:: .ASCIZ /%A ACTUAL OR MEASURED VALUE: %D3%N/
1696 004711 EF9006:: .ASCIZ /%A %T%A %D2%A(D)%N/
1697 004735 EF9010:: .ASCIZ /%A NUMBER OF ERRORS DETECTED ON LINE %D2%A(D) IS %D5%A(D)%N/
1698 005034 EF9016:: .ASCIZ /%A UNEXPECTED %T%A FOR LINE %D2%A(D) IN FIFO AFTER RESET:%N/
1699 005131 EF9017:: .ASCIZ /%A %T%A (WITH ERROR FLAGS) IS %D6%A(0)%N/
1700 005205 EF9018:: .ASCIZ /%A %T%A IN SELFTEST CODE FIFO SLOT FOR LINE %D2/
1701 005265 .ASCIZ /%A(D) AFTER RESET.%N/
1702 005312 EF9301:: .ASCIZ /%A %T%D2%A(D), BMP CODE REPORTED :%D3%A(0)%N/
1703 005370 EF9302:: .ASCIZ /%A OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)%N/
1704
1705 ;***** ERROR MESSAGES *****
1706 005470 DR00MG:: .ASCIZ /CSR/
1707 005474 DR02MG:: .ASCIZ /RBUF/
1708 005501 DR04MG:: .ASCIZ /LPR/
1709 005505 DR06MG:: .ASCIZ /FIFOSIZE,STAT/
1710 005523 DR10MG:: .ASCIZ /LNCTRL/
1711 005532 DR12MG:: .ASCIZ /TBUFFAD1/
1712 005543 DR14MG:: .ASCIZ /TBUFFAD2/
1713 005554 DR16MG:: .ASCIZ /TBUFFCT/
1714 005564 EM0103:: .ASCIZ / D' ICE REGISTER ADDRESS TEST FAILED./
1715 005632 EM0201:: .ASCIZ / MASTER RESET TEST FAILED./
1716 005665 EM0202:: .ASCIZ / MASTER RESET BIT DID NOT CLEAR AFTER BOARD RESET./
1717 005751 .ASCIZ / WAITED 5 SECONDS. BIT DEFECTIVE OR FIRMWARE HUNG./
1718 006040 EM0203:: .ASCIZ / MASTER RESET BIT CLEAR IMMEDIATELY AFTER BOARD RESET./
1719 006130 .ASCIZ / BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
1720 006203 EM0204:: .ASCIZ \ MR BIT WENT CLEAR WITHIN 1/2 SECOND OF BOARD RESET.\
1721 006271 .ASCIZ / BIT DEFECTIVE OR SELFTEST WAS (INCORRECTLY) SKIPPED./
1722 006362 EM0301:: .ASCIZ /MASTER RESET (SKIP SELFTEST) TEST FAILED./
1723 006434 EM0302:: .ASCIZ / MR BIT CLR WITHIN 10 MILLISECOND AFTER BOARD RESET./
1724 006521 .ASCIZ / BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
1725 006574 EM0303:: .ASCIZ \ MR BIT WENT CLEAR 1/5 TO 5 SECONDS AFTER RESET.\
1726 006656 .ASCIZ / SELFTEST DID NOT GET SKIPPED (SHOULD HAVE BEEN SKIPPED)./
1727 006753 EM0401:: .ASCIZ /RBUF REGISTER RX CHARACTER FIELD TEST FAILED./
1728 007031 EM0402:: .ASCIZ / IMPROPER CODE FOUND IN RX FIFO AFTER DUT RESET./
1729 007113 .ASCIZ / EXPECTED: SELFTEST CODE, ACTUAL: IMPROPER CODE./
1730 007201 EM0501:: .ASCIZ /RBUF REGISTER ERROR FLAGS FIELD TEST FAILED/
1731 007255 EM0502:: .ASCIZ / RX ERROR FLAG(S) FOUND CLEAR ON SELFTEST CODE./

```

```

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

```

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

1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830

```
;*****  
;  
;          FVTSKL2.P11  
;  
;*****  
  
.SBTTL  GLOBAL ERROR REPORT SECTION  
  
;**  
; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS  
; USED BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION. PRINTB  
; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.  
;--
```

1832 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0101 -
1833 ;*****
1834 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1835 ;* INFORMATION IF AN ERROR IS DETECTED IN TEST 1 (REGISTER ADDRESS
1836 ;* ACCESS TEST). IF THE "EXTENDED ERROR INFO" OPTION HAS BEEN SELECTED
1837 ;* THEN THIS SUBROUTINE WILL REPORT THE TYPE OF ACCESS (READ OR WRITE OR
1838 ;* BOTH) WHICH CAUSED A BUS TIME-OUT TRAP (004 TRAP). A MESSAGE INDICATING
1839 ;* THAT THE DHU MAY BE AT THE WRONG UNIBUS ADDRESS IS ALSO PRINTED.
1840 ;*
1841 ;* INPUTS: R5 - ERROR FLAG WORD.
1842 ;* IF BIT 0 IS SET, A READ ERROR OCCURED.
1843 ;* IF BIT 1 IS SET, A WRITE ERROR OCCURED.
1844 ;*
1845 ;* OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE.
1846 ;*
1847 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER0101" AS THE MESSAGE POINTER
1848 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
1849 ;*
1850 ;* COMMENTS:
1851 ;*
1852 ;* SUBORDINATE ROUTINES USED: NONE.
1853 ;*****
1854
1855 015444 BGNMSG ER0101
1856 015444 ER0101::
015444 SAVE ;SAVE THE GPR CONTENTS.
004567 166322 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
1857
1858 015450 012700 000100 MOV #BIT06,R0 ;SET-UP THE BIT MAP FOR 'REPORT EXT'D ERROR INFO'
1859 015454 046700 164540 BIC OPTION,R0 ;TRY AND CLEAR THE FLAG.
1860 015460 001036 BNE 64 ;EXIT IF OPTION NOT SELECTED.
1861 ;*
1862 ;* REPORT EXTENDED ERROR INFORMATION
1863 ;*
1864
1865 015462 032705 000001 BIT #BIT0,R5 ;TEST FOR READ ERROR.
1866 015466 001410 BEQ 24 ;SKIP READ ERROR MSG IF NO READ ERROR.
1867 015470 PRINTB #MSG1 ;PRINT READ ERROR MESSAGE.
015470 012746 015562 MOV #MSG1,-(SP)
015474 012746 000001 MOV #1,-(SP)
015500 010600 MOV SP,R0
015502 104414 TRAP C:PNTB
015504 062706 000004 ADD #4,SP
1868 015510 032705 000002 24: BIT #BIT1,R5 ;TEST FOR WRITE ERROR.
1869 015514 001410 BEQ 44 ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
1870 015516 PRINTB #MSG2 ;PRINT WRITE ERROR MESSAGE.
015516 012746 015640 MOV #MSG2,-(SP)
015522 012746 000001 MOV #1,-(SP)
015526 010600 MOV SP,R0
015530 104414 TRAP C:PNTB
015532 062706 000004 ADD #4,SP
1871 015536 44: PRINTX #MSG3 ;SUGGEST THAT DHU MAY BE AT WRONG ADDRESS.
015536 012746 015717 MOV #MSG3,-(SP)
015542 012746 000001 MOV #1,-(SP)
015546 010600 MOV SP,R0
015550 104415 TRAP C:PNTX
015552 062706 000004 ADD #4,SP

```

1872 015556          64: PASS          ;RESTORE THE GPR CONTENTS.
      015556 004736          JSR      PC,8(SP)+          ;RETURN TO PREG05 SUBRT.
1873 015560          ENDMSG
      015560          L10002: TRAP      C#MSG
      015560 104423
1874
1875 015562      045      101      102 MSG1:: .ASCIZ /#ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.#N/
      015565      125      123      040
      015570      124      111      115
      015573      105      055      117
      015576      125      124      040
      015601      124      122      101
      015604      120      040      103
      015607      101      125      123
      015612      105      104      040
      015615      102      131      040
      015620      122      105      101
      015623      104      040      101
      015626      124      124      105
      015631      115      120      124
      015634      056      045      116
      015637      000
1876 015640      045      101      102 MSG2:: .ASCIZ /#ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.#N/
      015643      125      123      040
      015646      124      111      115
      015651      105      055      117
      015654      125      124      040
      015657      124      122      101
      015662      120      040      103
      015665      101      125      123
      015670      105      104      040
      015673      102      131      040
      015676      127      122      111
      015701      124      105      040
      015704      101      124      124
      015707      105      115      120
      015712      124      056      045
      015715      116      000
1877 015717      045      101      104 MSG3:: .ASCIZ /#ADHU MAY BE AT THE WRONG UNIBUS ADDRESS.#N#N/
      015722      110      125      040
      015725      115      101      131
      015730      040      102      105
      015733      040      101      124
      015736      040      124      110
      015741      105      040      127
      015744      122      117      116
      015747      107      040      125
      015752      116      111      102
      015755      125      123      040
      015760      101      104      104
      015763      122      105      123
      015766      123      056      045
      015771      116      045      116
      015774      000
1878
1879          .EVEN

```

```

1881 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0201 -
1882 ;*****
1883 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS 2 CONTIGUOUS
1884 ;* ASCII ERROR MESSAGES. THE ADDRESS OF THE FIRST MESSAGE IS PASSED
1885 ;* AS AN INPUT PARAMETER AND THE ADDRESS OF THE SECOND IS FOUND BY
1886 ;* SEARCHING FOR THE END OF THE FIRST MESSAGE. THE MESSAGES ARE ONLY
1887 ;* PRINTED IF EXT'D ERROR REPORTING HAS BEEN REQUESTED.
1888 ;*
1889 ;* INPUTS: R1 - ADDRESS OF THE FIRST MESSAGE TO PRINT.
1890 ;*
1891 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
1892 ;*
1893 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE FIRST MESSAGE IN R1.
1894 ;* INCLUDE THE LABEL "ER0201" AS THE MESSAGE POINTER
1895 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1896 ;*
1897 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
1898 ;* THE SECOND MESSAGE SHOULD FOLLOW THE FIRST ONE IN THE PROGRAM
1899 ;* MEMORY. EACH MESSAGE SHOULD BE DEFINED USING .ASCIZ
1900 ;*
1901 ;* SUBORDINATE ROUTINES USED: NONE.
1902 ;*****
1903
1904 BGNMSG ER0201
1905 015776 015776 004567 165770
1906 015776 015776
1907 015776 004567 165770
1908 015776 004567 165770
1909 015776 004567 165770
1910 015776 004567 165770
1911 015776 004567 165770
1912 015776 004567 165770
1913 015776 004567 165770
1914 015776 004567 165770
1915 015776 004567 165770
1916 015776 004567 165770
1917 015776 004567 165770
1918 015776 004567 165770
1919 015776 004567 165770
1920 015776 004567 165770

```

ER0201::

```

;SAVE THE GPR CONTENTS.
R5,PREG05 ;CALL REGISTER SAVE SUBRT.

MOV #BIT06,R0 ;TRY TO CLEAR THE
BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
BNE 44 ;EXIT IF FLAG NOT SET.

24: MOV R1,R2
TSTB (R2)+ ;CHECK FOR A ZERO BYTE (END OF MESSAGE).
BNE 24 ;LOOP UNTIL NEXT MESSAGE IS FOUND.

PRINTB #EF0503,R1 ;PRINT THE FIRST MESSAGE.

MOV R1,-(SP)
MOV #EF0503,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTB
ADD #6,SP

1916 PRINTB #EF0503,R2 ;PRINT THE SECOND MESSAGE.

MOV R2,-(SP)
MOV #EF0503,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTB
ADD #6,SP

44: PASS ;RESTORE THE GPR CONTENTS.
JSR PC,B(SP)+ ;RETURN TO PREG05 SUBRT.

ENDMSG

L10003: TRAP C#MSG

```

```

1922 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0503 -
1923 ;*****
1924 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
1925 ;* MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER, PROVIDED
1926 ;* EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
1927 ;*
1928 ;* INPUTS: R1 - ADDRESS OF THE MESSAGE TO PRINT.
1929 ;*
1930 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
1931 ;*
1932 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE MESSAGE IN R1.
1933 ;* INCLUDE THE LABEL "ER0503" AS THE MESSAGE POINTER
1934 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1935 ;*
1936 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
1937 ;*
1938 ;* SUBORDINATE ROUTINES USED: NONE.
1939 ;*****
1940
1941 016072 BGNMSG ER0503
1942 016072
1943 016072 012700 000100
1944 016076 046700 164116
1945 016102 001011
1946
1947
1948 016104 PRINTB #EF0503,R1 ;PRINT THE MESSAGE.
1949 016104 010146
1950 016106 012746 004121
1951 016112 012746 000002
1952 016116 010600
1953 016120 104414
1954 016122 062706 000006
1955
1956 016126 21: ENDMSG
1957 016126
1958 016126 104423
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000

```

ER0503::

```

MOV R1,-(SP)
MOV #EF0503,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTB
ADD #6,SP

```

L10004: TRAP C#MSG

```
1952 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0504 -
1953 ;*****
1954 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1955 ;* MESSAGES WHEN ILLEGAL INTERRUPTS ARE RECEIVED.
1956 ;*
1957 ;* INPUTS: R1 - ADDRESS OF THE MESSAGE TO PRINT.
1958 ;* R2 - NUMBER OF ILLEGAL INTERRUPTS RECEIVED.
1959 ;*
1960 ;* OUTPUTS: MESSAGESS ARE PRINTED AT THE OPERATOR CONSOLE.
1961 ;*
1962 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE MESSAGE IN R1.
1963 ;* LOAD THE NUMBER OF ILLEGAL INTS IN R2.
1964 ;* INCLUDE THE LABEL "ER0504" AS THE MESSAGE POINTER
1965 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1966 ;*
1967 ;* COMMENTS:
1968 ;*
1969 ;* SUBORDINATE ROUTINES USED: NONE.
1970 ;*****
1971
1972 016130 BGNMSG ER0504
1973 016130 ER0504::
1974 016130 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
1975 016134 046700 164060 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
1976 016140 001022 BNE 2$ ;EXIT IF FLAG NOT SET.
1977
1978 016142 PRINTB #EF0503,R1 ;PRINT THE FIRST LINE OF THE MESSAGE.
1979 016142 010146 MOV R1,-(SP)
016144 012746 004121 MOV #EF0503,-(SP)
016150 012746 000002 MOV #2,-(SP)
016154 010600 MOV SP,R0
016156 104414 TRAP C#PNTB
015160 062706 000006 ADD #6,SP
1979 016164 PRINTX #EF0505,R2 ;PRINT THE NUMBER OF INTS RECEIVED.
016164 010246 MOV R2,-(SP)
016166 012746 004126 MOV #EF0505,-(SP)
016172 012746 000002 MOV #2,-(SP)
016176 010600 MOV SP,R0
016200 104415 TRAP C#PNTX
016202 062706 000006 ADD #6,SP
1980
1981 016206 2$: ENDMSG
016206
016206 104423 L10005: TRAP C#MSG
```

```
1983 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER1401 -
1984 ;*****
1985 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1986 ;* INFORMATION (IF REQUESTED DURING THE SOFTWARE QUESTIONS) IF AN ERROR
1987 ;* IS DETECTED IN THE ROM VERSION TEST. THIS SUBROUTINE ANALYSES THE INPUT
1988 ;* PARAMETERS WHICH CONTAIN THE ROM VERSION NUMBERS FOR PROC_1 AND PROC_2
1989 ;* AND REPORTS THE APPROPRIATE ERROR MESSAGE TO THE OPERATOR.
1990 ;*
1991 ;* INPUTS: R1 - CONTAINS THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
1992 ;* R3 - CONTAINS THE ROM VERSION NUMBER OF PROC_1.
1993 ;* R4 - CONTAINS THE ROM VERSION NUMBER OF PROC_2.
1994 ;*
1995 ;* OUTPUTS: BASIC AND EXTENDED ERROR MESSAGES ARE REPORTED AT THE
1996 ;* OPERATORS CONSOLE.
1997 ;*
1998 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER1401" AS THE MESSAGE POINTER
1999 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
2000 ;*
2001 ;* COMMENTS:
2002 ;*
2003 ;* SUBORDINATE ROUTINES USED: NONE.
2004 ;*****
2005
2006 016210 BGNMSG ER1401
2007 016210 ER1401::
2008 016210 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
2009 016214 046700 164000 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2010 016220 001053 BNE 60$ ;EXIT IF FLAG NOT SET.
2011
2012 016222 PRINTB #EF0503,R1 ;REPORT THE ERROR MESSAGE PASSED IN.
2013 016222 010146 MOV R1,-(SP)
2014 016224 012746 004121 MOV #EF0503,-(SP)
2015 016230 012746 000002 MOV #2,-(SP)
2016 016234 010600 MOV SP,R0
2017 016236 104414 TRAP C$PNTB
2018 016240 062706 000006 ADD #6,SP
2019
2020 ;*
2021 ; DETERMINE WHICH ROM VERSION NUMBER(S) ARE MISSING.
2022 ;*
2023
2024 016244 012705 000143 MOV #99.,R5 ;GET INVALID ROM NUMBER.
2025 016250 012701 012123 MOV #EM1405,R1 ;SELECT PROC_1 MESSAGE.
2026 016254 012702 012151 MOV #EM1407,R2 ;SELECT THE "NOT FOUND" MESSAGE.
2027 016260 120305 CMPB R3,R5 ;CHECK PROC_1 ROM VERSION NUMBER.
2028 016262 001402 BEQ 2$ ;GO REPORT PROC_1 CODE NOT FOUND.
2029 016264 012702 012163 MOV #EM1408,R2 ;SELECT "FOUND" MESSAGE.
2030 016270 004767 000026 2$: JSR PC,50$ ;GO REPORT MESSAGE.
2031
2032 016274 012701 012136 MOV #EM1406,R1 ;SELECT PROC_2 MESSAGE.
2033 016300 012702 012151 MOV #EM1407,R2 ;SELECT THE "NOT FOUND" MESSAGE.
2034 016304 120405 CMPB R4,R5 ;CHECK PROC_2 ROM VERSION NUMBER.
2035 016306 001402 BEQ 4$ ;GO REPORT PROC_2 CODE NOT FOUND.
2036 016310 012702 012163 MOV #EM1408,R2 ;SELECT "FOUND" MESSAGE.
2037 016314 004767 000002 4$: JSR PC,50$ ;GO REPORT THE MESSAGE.
2038 016320 000413 BR 60$ ;EXIT.
```

```
2033
2034 016322      50$: PRINTX  #EF1402,R1,R2 ;REPORT THE MESSAGE.
      016322      010246
      016324      010146
      016326      012746      004303
      016332      012746      000003
      016336      010600
      016340      104415
      016342      062706      000010
2035 016346      000207
2036 016350      60$: RTS PC ;RETURN.
      016350
      016350      104423

MOV R2,-(SP)
MOV R1,-(SP)
MOV #EF1402,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #10,SP

L10006:
TRAP C#MSG
```

B⁵,

```
2038 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER1601 -
2039 *****
2040 ;* THIS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
2041 ;* INFORMATION IF AN ERROR IS DETECTED IN ONE OF THE DEVICE REGISTER
2042 ;* ACCESS TESTS, PROVIDED EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
2043 ;* THIS SUBROUTINE REPORTS THE ACTUAL AND EXPECTED FROM THE DEVICE
2044 ;* REGISTER(S) WHICH IS(ARE) IN FAULTY.
2045 ;*
2046 ;* INPUTS:      R1 - ACTUAL DATA (UNUSED BITS SET TO 0).
2047 ;*              R2 - EXPECTED DATA (UNUSED BITS SET TO 0).
2048 ;*              R3 - OFFSET (IN BYTES) TO THE REGISTER BEING TESTED.
2049 ;*              R5 - LINE NUMBER OF REGISTER BEING TESTED.
2050 ;*              RMATBB - LABEL AT BASE OF REGISTER MESSAGE ADDRESS TABLE.
2051 ;*
2052 ;* OUTPUTS:     MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
2053 ;*
2054 ;* CALLING SEQUENCE:  INCLUDE THE LABEL "ER1601" AS THE MESSAGE POINTER
2055 ;*                   PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
2056 ;*
2057 ;* COMMENTS:
2058 ;*
2059 ;* SUBORDINATE ROUTINES CALLED: NONE
2060 ;* *****
2061 BGNMSG ER1601
2062
2063 016352 012 JO 000100      MOV  #BIT06,R0      ;TRY TO CLEAR THE
2064 016356 046700 163636    BIC  OPTION,R0      ;EXT'D ERROR REPORTING FLAG
2065 016362 001036          BNE  21              ;EXIT IF FLAG NOT SET.
2066
2067
2068 016364 016304 002256    MOV  RMATBB(R3),R4    ;FETCH ADDRESS OF REGISTER NAME MESSAGE.
2069
2070 016370          PRINTB  #EF1604,R4,R5          ;REPORT BASIC MESSAGE (REG NAME AND LINE #).
2071 016370 010546          MOV  R5,-(SP)
2072 016372 010446          MOV  R4,-(SP)
2073 016374 012746 004476    MOV  #EF1604,-(SP)
2074 016400 012746 000003    MOV  #3,-(SP)
2075 016404 010600          MOV  SP,R0
2076 016406 104414          TRAP  C:PNTB
2077 016410 062706 000010    ADD  #10,SP
2078
2079 016414          PRINTX  #EF1602,R2            ;PRINT THE EXPECTED DATA.
2080 016414 010246          MOV  R2,-(SP)
2081 016416 012746 004372    MOV  #EF1602,-(SP)
2082 016422 012746 000002    MOV  #2,-(SP)
2083 016426 010600          MOV  SP,R0
2084 016430 104415          TRAP  C:PNTX
2085 016432 062706 000006    ADD  #6,SP
2086
2087 016436          PRINTX  #EF1603,R1            ;PRINT THE ACTUAL DATA.
2088 016436 010146          MOV  R1,-(SP)
2089 016440 012746 004434    MOV  #EF1603,-(SP)
2090 016444 012746 000002    MOV  #2,-(SP)
2091 016450 010600          MOV  SP,R0
2092 016452 104415          TRAP  C:PNTX
2093 016454 062706 000006    ADD  #6,SP
2094
2095 21:      ENDMSG
2096
2097 L10007:
```

C5

CZDMUAD DMU 11 FUNC TST PART1 MACRO M1200 12 DEC 83 16:08 PAGE 42-1
GLOBAL ERROR REPORTING ROUTINE - ER1601 -

SEQ 54

016460 104423

TRAP CMSG

```
2075 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER1603 -
2076 ;*****
2077 ;* THIS ERROR REPORTING ROUTINE IS USED TO PRINT OUT A BASIC ERROR
2078 ;* MESSAGE, ALONG WITH A MESSAGE INFORMING THE OPERATOR WHICH TEST IS
2079 ;* ABOUT TO BE ABORTED, PROVIDED EXTENDED ERROR INFORMATION HAS BEEN
2080 ;* REQUESTED, OTHERWISE ONLY A "TEST FAILURE" MESSAGE WILL BE PRINTED.
2081 ;*
2082 ;* INPUTS: R1 - CONTAINS THE ADDRESS OF THE MESSAGE TO BE PRINTED.
2083 ;* ERRMSG - CONTAINS THE ADDRESS OF THE MESSAGE THAT INDICATES
2084 ;* THE TEST THAT IS BEING PERFORMED, EG DMA, BREAK ETC.
2085 ;*
2086 ;* OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
2087 ;* "TESTNAME TEST ABORTED"
2088 ;*
2089 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER1603" AS THE MESSAGE POINTER
2090 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
2091 ;*
2092 ;* COMMENTS:
2093 ;*
2094 ;*
2095 ;* SUBORDINATE ROUTINES CALLED: NONE.
2096 ;*****
2097 016462 BGNMSG ER1603
2098 016462 ER1603::
016462 004567 165304 SAVE JSR R5,PREG05 ;SAVE THE CONTENTS OF THE GPRS.
;CALL REGISTER SAVE SUBRT.
2099
2100 016466 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
2101 016472 046700 163522 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2102 016476 001024 BNE 21 ;EXIT IF FLAG NOT SET.
2103
2104
2105 016500 PRINTB #EF0503,R1 ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
016500 010146 MOV R1,-(SP)
016502 012746 004121 MOV #EF0503,-(SP)
016506 012746 000002 MOV #2,-(SP)
016512 010600 MOV SP,R0
016514 104414 TRAP C:PNTB
016516 062706 000006 ADD #6,SP
2106
2107 016522 016702 165240 MOV ERRMSG,R2 ;GET THE "TEST MESSAGE".
2108 016526 PRINTB #EF1601,R2 ;PRINT "TEST ABORTED" MESSAGE.
016526 010246 MOV R2,-(SP)
016530 012746 004340 MOV #EF1601,-(SP)
016534 012746 000002 MOV #2,-(SP)
016540 010600 MOV SP,R0
016542 104414 TRAP C:PNTB
016544 062706 000006 ADD #6,SP
2109
2110 016550 21: PASS ;RESTORE THE CONTENTS OF THE GPRS.
016550 004736 JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
2111 016552 ENDMSG
016552 104423 L10010: TRAP C:MSG
```

```
2113 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER3001 -
2114 ;*****
2115 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS INTENDED FOR USE IN THE
2116 ;* INTERRUPT BR LEVEL TEST. IT REPORTS ADDITIONAL INFORMATION WHEN AN
2117 ;* INTERRUPT HAS OCCURRED AT THE WRONG BR LEVEL. UNLESS EXTENDED ERROR
2118 ;* REPORTING HAS BEEN REQUESTED, ONLY THE TEST FAIL MESSAGE
2119 ;* BE PRINTED.
2120 ;*
2121 ;* INPUTS: R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
2122 ;* R4 - BR LEVEL AT WHICH THE INT REQUEST OCCURRED.
2123 ;* R5 - EXPECTED OR CORRECT BR LEVEL FOR THE OUT.
2124 ;*
2125 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
2126 ;*
2127 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER3001" AS THE MESSAGE POINTER
2128 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2129 ;*
2130 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2131 ;*
2132 ;* SUBORDINATE ROUTINES USED: NONE.
2133 ;*****
2134
2135 016554 BGNMSG ER3001
2136 016554 ER3001::
2137 016554 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
2138 016560 046700 163434 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2139 016564 001033 BNE 21 ;EXIT IF FLAG NOT SET.
2140
2141 016566 PRINTB #EF0503,R1 ;PRINT THE FIRST LINE OF THE MESSAGE.
2142 016566 010146 MOV R1,-(SP)
2143 016570 012746 004121 MOV #EF0503,-(SP)
2144 016574 012746 000002 MOV #2,-(SP)
2145 016600 010600 MOV SP,R0
2146 016602 104414 TRAP C#PNTB
2147 016604 062706 000006 ADD #6,SP
2148 016610 PRINTX #EF3001,R5 ;REPORT EXPECTED BR LEVEL.
2149 016610 010546 MOV R5,-(SP)
2150 016612 012746 004573 MOV #EF3001,-(SP)
2151 016616 012746 000002 MOV #2,-(SP)
2152 016622 010600 MOV SP,R0
2153 016624 104415 TRAP C#PNTX
2154 016626 062706 000006 ADD #6,SP
2155 016632 PRINTX #EF3002,R4 ;REPORT ACTUAL BR LEVEL.
2156 016632 010446 MOV R4,-(SP)
2157 016634 012746 004642 MOV #EF3002,-(SP)
2158 016640 012746 000002 MOV #2,-(SP)
2159 016644 010600 MOV SP,R0
2160 016646 104415 TRAP C#PNTX
2161 016650 062706 000006 ADD #6,SP
2162
2163 21: ENDMSG
2164
2165 016654 L10011:
2166 016654 TRAP C#MSG
2167 016654 104423
```

```
2147 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9004 -
2148 ;*****
2149 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH REPORTS ERROR SUMMARIES
2150 ;* FOR LINES WHICH HAVE EXCEEDED THE SPECIFIED MAXIMUM NUMBER OF
2151 ;* INDIVIDUAL RECEPTION ERRORS, PROVIDED EXTENDED ERROR REPORTING HAS
2152 ;* BEEN REQUESTED BY THE OPERATOR.
2153 ;*
2154 ;* INPUTS: R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
2155 ;* ERCNTB - LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
2156 ;* ERSRFR - "REPORT ERROR SUMMARY FOR LINE" FLAGS.
2157 ;*
2158 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2159 ;*
2160 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9004" AS THE MESSAGE POINTER
2161 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2162 ;*
2163 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2164 ;* THE CONTENTS OF GPR'S R2, R3, R4, AND R5 ARE DESTROYED.
2165 ;*
2166 ;* SUBORDINATE ROUTINES USED: NONE.
2167 ;*****
2168
2169 016656 BGNMSG ER9004
2170 016656 ER9004::
2171 016656 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
2172 016662 046700 163332 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2173 016666 001040 BNE 6# ;EXIT IF FLAG NOT SET.
2174
2175 016670 PRINTB #EF0503,#EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
2176 016670 012746 014716 MOV #EM9014,-(SP)
2177 016674 012746 004121 MOV #EF0503,-(SP)
2178 016700 012746 000002 MOV #2,-(SP)
2179 016704 010600 MOV SP,R0
2180 016706 104414 TRAP C#PNTB
2181 016710 062706 000006 ADD #6,SP
2182 016714 005002 CLR R2 ;CLEAR THE LINE COUNTER.
2183 016716 016703 163534 MOV ERSRFR,R3 ;GET THE ERROR SUMMARY FLAGS.
2184 016722 005004 CLR R4 ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
2185 016724 000241 2# CLC ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
2186 016726 006003 ROR R3 ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
2187 016730 103013 BCC 4# ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
2188 016732 PRINTX #EF9010,R2,ERCNTB(R4)
2189 016732 016446 002460 MOV ERCNTB(R4),-(SP)
2190 016736 010246 MOV R2,-(SP)
2191 016740 012746 004735 MOV #EF9010,-(SP)
2192 016744 012746 000003 MOV #3,-(SP)
2193 016750 010600 MOV SP,R0
2194 016752 104415 TRAP C#PNTX
2195 016754 062706 000010 ADD #10,SP
2196 016760 012405 4# MOV (R4)+,R5 ;INCREMENT THE LINE OFFSET BY 2.
2197 016762 005202 INC R2 ;INCREMENT THE LINE COUNTER.
2198 016764 005703 TST R3 ;CHECK THE ERROR SUMMARY FLAGS.
2199 016766 001356 BNE 2# ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.
2200
2201 6# ENDMMSG
2202
2203 016770 L10012:
2204 016770
```

G5

CZDHUAO DHU 11 FUNC TST PART1 MACRO M1200 12-DEC 83 16:08 PAGE 45-1
GLOBAL ERROR REPORTING ROUTINE - ER9004 -

SEQ 58

016770 104423

TRAP C\$MSG

```

2190 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9007 -
2191 ;*****
2192 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS USED TO REPORT THAT
2193 ;* SOMETHING OTHER THAN A SELFTEST CODE WAS FOUND IN A SELFTEST CODE
2194 ;* FIFO SLOT DURING THE REMOVAL OF THE SELFTEST CODES FROM THE FIFO.
2195 ;* THIS ROUTINE IS USED BY THE RSTRT ROUTINE. EXTENDED ERROR INFORMATION
2196 ;* IS GIVEN ONLY WHEN REQUESTED IN THE SOFTWARE QUESTIONS.
2197 ;*
2198 ;* INPUTS: R1 - ADDRESS OF ERROR MESSAGE QUALIFIER STRING.
2199 ;* R2 - INCORRECT CODE AS READ FROM THE SELFTEST CODE FIFO SLOT.
2200 ;* R3 - LINE NUMBER ASSOCIATED WITH THE SELFTEST FIFO SLOT.
2201 ;*
2202 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2203 ;*
2204 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9007" AS THE MESSAGE POINTER
2205 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2206 ;*
2207 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2208 ;*
2209 ;* SUBORDINATE ROUTINES USED: NONE.
2210 ;*****
2211
2212 016772 BGNMSG ER9007
2213 016772
2214 016772 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
2215 016776 046700 163216 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2216 017002 001026 BNE 24 ;EXIT IF FLAG NOT SET.
2217
2218 017004 042703 177760 BIC #177760,R3 ;REMOVE ALL BUT LINE # BITS FROM LINE # WORD.
2219 017010 PRINTB #EF9018,R1,R3 ;REPORT SECONDARY ERROR MESSAGE.
2220 017010 010346 MOV R3,-(SP)
2221 017012 010146 MOV R1,-(SP)
2222 017014 012746 005205 MOV #EF9018,-(SP)
2223 017020 012746 000003 MOV #3,-(SP)
2224 017024 010600 MOV SP,R0
2225 017026 104414 TRAP C#PNTB
2226 017030 062706 000010 ADD #10,SP
2227 017034 PRINTX #EF9017,R1,R2 ;REPORT THE ACTUAL INCORRECT CODE.
2228 017036 010146 MOV R2,-(SP)
2229 017040 012746 005131 MOV R1,-(SP)
2230 017044 012746 000003 MOV #EF9017,-(SP)
2231 017050 010600 MOV #3,-(SP)
2232 017052 104415 MOV SP,R0
2233 017054 062706 000010 TRAP C#PNTX
2234 ADD #10,SP
2235
2236 24: ENDMSG
2237
2238 L10013:
2239 TRAP C#MSG

```

```

2224 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9008 -
2225 ;*****
2226 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS USED TO REPORT THAT
2227 ;* AN UNEXPECTED CODE OR CHARACTER HAS BEEN FOUND IN THE DUT RECEIVE
2228 ;* CHARACTER FIFO. THE ADDITIONAL ERROR IS REPORTED ONLY IF REQUESTED
2229 ;* DURING THE SOFTWARE QUESTIONS.
2230 ;*
2231 ;* INPUTS: R1 - ADDRESS OF PARTIAL ERROR MESSAGE STRING.
2232 ;* R2 - INCORRECT CODE AS READ FROM THE SELFTEST CODE FIFO SLOT.
2233 ;*
2234 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2235 ;*
2236 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9008" AS THE MESSAGE POINTER
2237 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2238 ;*
2239 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2240 ;*
2241 ;* SUBORDINATE ROUTINES USED: NONE.
2242 ;*****
2243
2244 017062 BGNMSG ER9008
2245 017062 ER9008::
2246 017062 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
2247 017066 046700 163126 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2248 017072 001030 BNE 21 ;EXIT IF FLAG NOT SET.
2249
2250 ;*
2251 ;* EXTRACT THE LINE NUMBER FROM THE INCORRECT CODE OR CHARACTER WHICH WAS READ
2252 ;* FROM THE SELFTEST CODE FIFO SLOT.
2253 ;*
2254 017074 010203 MOV R2,R3
2255 017076 000303 SWAB R3
2256 017100 042703 177760 BIC #177760,R3 ;CALCULATE LINE NUMBER OF CODE.
2257 017104 PRINTB #EF9016,R1,R3 ;REPORT TYPE OF INCORRECT CODE FOUND.
2258 017104 010346 MOV R3,-(SP)
2259 017106 010146 MOV R1,-(SP)
2260 017110 012746 005034 MOV #EF9016,-(SP)
2261 017114 012746 000003 MOV #3,-(SP)
2262 017120 010600 MOV SP,R0
2263 017122 104414 TRAP C#PNTB
2264 017124 062706 000010 ADD #10,SP
2265 017130 PRINTX #EF9017,R1,R2 ;REPORT THE ACTUAL INCORRECT CODE.
2266 017130 010246 MOV R2,-(SP)
2267 017132 010146 MOV R1,-(SP)
2268 017134 012746 005131 MOV #EF9017,-(SP)
2269 017140 012746 000003 MOV #3,-(SP)
2270 017144 010600 MOV SP,R0
2271 017146 104415 TRAP C#PNTX
2272 017150 062706 000010 ADD #10,SP
2273
2274 21: ENDMSG
2275
2276 017154 L10014:
2277 017154 TRAP C#MSG
2278 017154 104423

```

```

2262 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9101 -
2263 ;*****
2264 ;* THIS IS A GENERAL ERROR REPORTING SUBROUTINE WHICH REPORTS A MESSAGE
2265 ;* WHICH TAKES A SINGLE, 2 DIGIT DECIMAL ARGUMENT AFTER THE END OF AN
2266 ;* ASCII MESSAGE.
2267 ;*
2268 ;* INPUTS: R1 - VALUE TO BE PRINTED AFTER MSG AS 2 DECIMAL DIGITS.
2269 ;* R2 - ADDRESS OF MESSAGE TO PRINT FIRST.
2270 ;*
2271 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
2272 ;*
2273 ;* CALLING SEQUENCE INCLUDE THE LABEL "ER9101" AS THE MESSAGE POINTER
2274 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2275 ;*
2276 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2277 ;*
2278 ;* SUBORDINATE ROUTINES USED: NONE.
2279 ;*****
2280
2281 017156 BGNMSG ER9101
2282 017156 ER9101::
2283 017156 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
2284 017162 046700 163032 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2285 017166 001012 BNE 24 ;EXIT IF FLAG NOT SET.
2286
2287
2288 017170 PRINTB #EF9006,R2,R1 ;REPORT THE STRING FOLLOWED BY THE NUMBER.
2289 017170 010146 MOV R1,-(SP)
2290 017172 010246 MOV R2,-(SP)
2291 017174 012746 004711 MOV #EF9006,-(SP)
2292 017200 012746 000003 MOV #3,-(SP)
2293 017204 010600 MOV SP,R0
2294 017206 104414 TRAP C#PNTB
2295 017210 062706 000010 ADD #10,SP
2296
2297 24: ENDMSG
2298
2299 017214 L10015:
2300 017214 104423 TRAP C#MSG

```

```

2292 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9301 -
2293 ;*****
2294 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ANY BMP CODES
2295 ;* THAT ARE FOUND IN THE BMP CODE QUEUE, TOGETHER WITH THE NUMBER OF
2296 ;* THE TEST THAT WAS EXECUTING AT THE TIME THE BMP CODE WAS LOGGED.
2297 ;* PROVIDED EXTENDED ERROR REPORTING HAS BEEN ENABLED.
2298 ;*
2299 ;* INPUTS: R1 - THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
2300 ;* R2 - THE ADDRESS OF THE NEXT EMPTY CELL IN THE QUEUE.
2301 ;*
2302 ;* OUTPUTS: THE TEST NUMBER FOLLOWED BY THE BMP CODE ARE PRINTED AT THE
2303 ;* OPERATOR CONSOLE.
2304 ;*
2305 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9301" AS THE MESSAGE POINTER
2306 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2307 ;*
2308 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2309 ;*
2310 ;* SUBORDINATE ROUTINES USED: NONE.
2311 ;*****
2312
2313 BGNMSG ER9301
2314 017216 017216 ER9301::
2315 017216 004567 164550 SAVE JSR ;SAVE THE GPRS ON THE STACK.
2316 017222 012700 000100 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2317 017226 046700 162766 MOV #BIT06,R0 ;TRY TO CLEAR THE
2318 017232 001064 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2319 BNE 60$ ;EX' IF FLAG NOT SET.
2320 017234 PRINTB #EF0503,R1 ;REPORT UNEXPECTED BMP CODES FOUND.
2321 017234 010146 MOV R1,-(SP)
2322 017236 012746 004121 MOV #EF0503,-(SP)
2323 017242 012746 000002 MOV #2,-(SP)
2324 017246 010600 MOV SP,R0
2325 017250 104414 TRAP C#PNTB
2326 017252 062706 000006 ADD #6,SP
2327 017256 012703 002522 MOV #BMPQ08,R3 ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
2328 017262 012705 015273 MOV #EM9302,R5 ;GET THE MESSAGE TO BE REPORTED.
2329 2$: MOV (R3)+,R1 ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
2330 MOV (R3)+,R4 ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
2331 JSR PC,50$ ;GO REPORT THE BMP CODE.
2332 CMP R3,R2 ;CHECK IF ALL CODES HAVE BEEN REPORTED.
2333 BLO 2$ ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
2334 ;*
2335 ;* CHECK IF OVERFLOW HAS OCCURRED.
2336 ;* THE CONDITIONS FOR OVERFLOW ARE: THE POINTER CONTAINS THE ADDRESS OF THE
2337 ;* LAST CELL IN THE QUEUE, AND A BMP CODE HAS ALREADY BEEN WRITTEN INTO THAT
2338 ;* CELL.
2339 ;*
2340 2334 017302 020227 002716 CMP R2,#BMPQ0E-4 ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
2341 2335 017306 001036 BNE 60$ ;EXIT IF NOT AT THE LAST LOCATION.
2342 2336 017310 005762 000002 TST 2(R2) ;CHECK FOR A BMP CODE IN THE LAST CELL
2343 2337 017314 001433 BEQ 60$ ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
2344 2338 017316 012301 MOV (R3)+,R1 ;GET THE TEST NUMBER OFF THE QUEUE.
2345 2339 017320 011304 MOV (R3),R4 ;GET THE BMP CODE OFF THE QUEUE.
2346 2340 017322 012705 015323 MOV #EM9303,R5 ;SELECT THE MESSAGE TO BE REPORTED.

```

```

2341 017326          PRINTX  #EF9302          ;REPORT OVERFLOW CONDITION.
      017326 012746 005370
      017332 012746 000001
      017336 010600
      017340 104415
      017342 062706 000004
2342 017346 004767 000002
2343 017352 000414
2344
2345 017354          50$: PRINTX  #EF9301,R5,R1,R4 ;PRINT THE MESSAGE.
      017354 010446
      017356 010146
      017360 010546
      017362 012746 005312
      017366 012746 000004
      017372 010600
      017374 104415
      017376 062706 000012
2346 017402 000207
2347 017404
      017404 004736
2348
2349 017406          60$: RTS      PC          ;RETURN.
      017406
      017406 104423          PASS          ;RESTORE THE GPR CONTENTS.
                                   JSR      PC,@(SP)+ ;RETURN TO PREG05 SUBRT.

                                   ENDMSG

                                   L10016:
                                   TRAP     C#MSG

```

```
2351 .SBTTL GLOBAL SUBROUTINES SECTION
2353 ;*****
2354 ;
2355 ; FVTSKL3.P11
2356 ;
2357 ;*****
2359
2360
2361 ;++
2362 ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2363 ; THAT ARE USED IN MORE THAN ONE TEST.
2364 ;--
```

```
2366 .SBTTL GLOBAL SUBROUTINE - ALTFLD -
2367 ;* *****
2368 ;* - ALTER DEVICE REGISTER FIELDS ROUTINE -
2369 ;* THIS SUBROUTINE ALTERS THE SPECIFIED FIELD OF THE SPECIFIED DEVICE
2370 ;* REGISTER FOR THE SPECIFIED LINES. THIS ROUTINE CAN BE USED TO SET
2371 ;* OR CLEAR BITS WITHIN SELECTED FIELDS OF SELECTED REGISTERS.
2372 ;* USE EXAMPLES: SET RX.BAUD.RATE FIELDS ON LINES 3 AND 6.
2373 ;* CLEAR TX.DMA BITS ON ALL LINES.
2374 ;*
2375 ;* INPUTS: R1 - ADDRESS OF THE REGISTERS TO ALTER.
2376 ;* R2 - BIT FIELDS SET TO DESIRED STATES.
2377 ;* R3 - BIT MAP OF LINES FOR WHICH TO ALTER REGISTER.
2378 ;* R4 - MASK OF BITS TO ALTER (1 INDICATES CHANGE BIT).
2379 ;* CSRA - CONTAINS THE ADDRESS OF THE DEVICE CSR.
2380 ;* IESTAT - SAVED STATES OF THE INTERRUPT ENABLE BITS.
2381 ;*
2382 ;* OUTPUTS: DEVICE REGISTERS - SPECIFIED REGISTER FIELDS ALTERED.
2383 ;* CSR IND.ADR.REG FIELD - DESTROYED.
2384 ;*
2385 ;* CALLING SEQUENCE: JSR PC,ALTFLD
2386 ;*
2387 ;* COMMENTS: THIS ROUTINE READS THE SPECIFIED REGISTERS FOR ALL LINES
2388 ;* WITH NUMBERS LOWER THAN THE HIGHEST SPECIFIED LINE.
2389 ;* THIS ROUTINE DOES NOT READ THE CSR.
2390 ;*
2391 ;* SUBROUTINES CALLED: NONE.
2392 ;* -- *****
2393
2394 017410 004567 164356 ALTFLD:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
2395 017410 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2396
2397 ;*
2398 ;* SET UP TO LOOP FOR EACH LINE:
2399 ;* PREPARE THE WORD TO BE ORED INTO THE REGISTER CONTENTS.
2400 ;* SET UP THE WORD TO WRITE INTO THE IND.ADR.REG FIELD OF THE CSR.
2401 ;*
2402 017414 010400 MOV R4,R0 ;CALCULATE THE NEW CONTENTS OF THE
2403 017416 005100 COM R0 ; REGISTER FIELDS WHICH ARE TO BE
2404 017420 040002 BIC R0,R2 ; ALTERED BY THIS ROUTINE.
2405 017422 016705 162656 MOV IESTAT,R5 ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.
2406
2407 ;*
2408 ;* LOOP ONCE FOR EACH LINE, ALTERING THE SPECIFIED FIELD IN THE SPECIFIED
2409 ;* REGISTER IF THE LINE HAS BEEN SELECTED FOR ALTERING.
2410 ;* EXIT THE LOOP IF NO MORE LINES TO ALTER, OR IF WE HAVE ALTERED THE MAX
2411 ;* ALLOWABLE NUMBER OF LINES (AS SPECIFIED BY NUMLNS).
2412 ;*
2413 017426 000241 CLC ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
2414 017430 006003 24: ROR R3 ;GET THE LINE SELECT BIT FOR THIS LINE.
2415 017432 103006 BCC 44 ;SKIP SETUP IF LINE IS NOT SELECTED.
2416 017434 010577 162576 MOV R5,BCSRA ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
2417 017440 011100 MOV (R1),R0 ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
2418 017442 040400 BIC R4,R0 ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
2419 017444 050200 BIS R2,R0 ;OR IN THE NEW STATES OF THE FIELDS.
2420 017446 010011 MOV R0,(R1) ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
2421 017450 005205 44: INC R5 ;SET LINE NUMBER TO THE NEXT LINE.
2422 017452 005703 TST R3 ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
2423 017454 001365 BNE 24 ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.
```

B6

CZDHUAO DHU 11 FUNC TST PART1 MACRO M1200 12 DEC-83 16:08 PAGE 51 1
GLOBAL SUBROUTINE ALTFLD -

SEQ 66

2422
2423 017456 601: PASS JSR ;RESTORE GPRS.
017456 004736 PC,B(SP);
2424 017460 000207 RTS PC ;RETURN TO PREG05 SUBRT.
;RETURN TO CALLING ROUTNE.

Cf,

```
2426 .SBTTL GLOBAL SUBROUTINE - CALMSL -
2427 ;* *****
2428 ;* - CALIBRATE MILLI SECOND LOOP COUNT SUBROUTINE -
2429 ;* THIS SUBROUTINE CALIBRATES THE TIMING LOOP WHICH IS USED IN THE MSLOOP
2430 ;* ROUTINE. THIS SUBROUTINE CALCULATES A VALUE FOR THE MSLCNT VARIABLE
2431 ;* WHICH IS THE NUMBER OF SOFTWARE LOOPS WHICH TAKES 1 MS TO EXECUTE IN
2432 ;* THE MSLOOP ROUTINE. THIS ROUTINE CALIBRATES THE COUNT BY USING THE
2433 ;* LINE TIME CLOCK (LTC), SO IF NO LTC IS AVAILABLE THE DEFAULT VALUE FOR
2434 ;* THE DELAY COUNT MUST BE USED.
2435 ;*
2436 ;*
2437 ;* INPUTS: MSLCNT - DEFAULT 1 MS DELAY LOOP COUNT VALUE, OR
2438 ;* VALUE FROM PREVIOUS CALIBRATION.
2439 ;* MSTICK - NUMBER OF MS PER LTC CLOCK TICK.
2440 ;* TIMER1 - TIMER COUNTER CHANGED BY LTC INTERRUPT SERVICE RTN.
2441 ;* CLKHRZ - NUMBER OF LTC CLICKS PER SECOND (50 OR 60).
2442 ;*
2443 ;* OUTPUTS: CARRY - SET IF LTC IS AVAILABLE, AND NEW CALIBRATION PERFORMED.
2444 ;* MSLCNT - NEW 1 MS DELAY LOOP COUNT VALUE IF LTC AVAILABLE, OR
2445 ;* UNCHANGED IF NO LTC IS AVAILABLE.
2446 ;*
2447 ;* CALLING SEQUENCE: JSR PC,CALMSL
2448 ;*
2449 ;* COMMENTS:
2450 ;*
2451 ;* SUBORDINATE ROUTINES CALLED: UNSDIV,OOPS.
2452 ;* --- *****
2453
2454 017462 CALMSL:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
017462 004567 164304 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2455 017466 005067 000210 CLR 62H ;CLEAR THE 2ND TIME FLAG.
2456
2457 ;* SYNCHRONIZE WITH THE LTC.
2458 ;*
2459 017472 012705 000001 2H: MOV #1,R5 ;SET OUTER LOOP COUNTER TO 1 LOOP.
2460 ;* INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE ***
2461 ;* FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. ***
2462 017476 005000 CLR R0 ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
2463 017500 012767 000001 162632 MOV #1,TIMER1 ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
2464 017506 005767 162626 4H: TST TIMER1 ;CHECK FOR COUNTER HAVING GONE TO ZERO.
2465 017512 001410 BEQ 6H ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
2466 017514 005200 INC R0 ;COUNT THIS ITERATION OF THE INNER LOOP.
2467 017516 001373 BNE 4H ;LOOP IF COUNTER HAS NOT TURNED OVER.
2468 017520 005305 DEC R5 ;DECREMENT THE INNER LOOP COUNTER.
2469 017522 003371 BGT 4H ;LOOP IF OUTER LOOP COUNT NOT UP.
2470
2471 ;* IF WE GOT NO LTC INTERRUPT, INDICATE THAT THERE IS NO LTC AVAILABLE.
2472 ;* LTC MUST BE FLAKEY, OR NOT REALLY AN LTC AT ALL.
2473 ;*
2474 017524 005067 162606 CLR CLKHRZ ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
2475 017530 000241 CLC ;INDICATE FAILURE FOR RETURN.
2476 017532 000461 BR 60H ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
2477
2478 ;* WE ARE NOW SYNCHRONIZED WITH THE LTC.
2479 ;* SET UP FOR THE CALIBRATION LOOP.
2480 ;*
2481 017534 012704 002340 6H: MOV #TIMER1,R4 ;WILL TEST TIMER1 IN THE LOOP BELOW.
```

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

```
2536 .SBTTL GLOBAL SUBROUTINE - CKTRAP -
2537 ;*****
2538 ;* CHECK TRAP ROUTINE -
2539 ;* THIS SUBROUTINE IS USED TO CHECK FOR A BUS TIME-OUT TRAP (004 TRAP)
2540 ;* WHICH IS CAUSED BY AN ACCESS TO A NON-EXISTENT MEMORY OR I/O LOCATION.
2541 ;* IF THE TRAP DOES NOT OCCUR, THIS ROUTINE RETURNS A SUCCESS INDICATION.
2542 ;*
2543 ;* INPUTS: R0 - SOURCE ADDRESS FOR MOVE.
2544 ;* R1 - DESTINATION ADDRESS FOR MOVE.
2545 ;* (R0) - SOURCE FOR THE MOVE.
2546 ;*
2547 ;* OUTPUTS: (R1) - WRITTEN TO THE CONTENTS OF (R0).
2548 ;* CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED.
2549 ;* TP4FLG - NONZERO IF TRAP OCCURRED, CLEARED OTHERWISE.
2550 ;*
2551 ;* CALLING SEQUENCE: JSR PC,CKTRAP
2552 ;*
2553 ;* COMMENTS: IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS WHICH
2554 ;* IS LABELED ADRPTR WILL BE THE TRAP PC ADDRESS ON THE STACK.
2555 ;*
2556 ;* SUBORDINATE ROUTINES CALLED: NONE.
2557 ;*****
2558
2559 017706 CKTRAP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2560 017706 004567 164060 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2561 017712 005067 162376 CLR TP4FLG ;CLEAR THE 004 TRAP FLAGS.
2562 017720 005767 162370 MOV (R0),(R1) ;PERFORM THE MOVE IN QUESTION.
2563 017724 000261 ADRPTR:: TST TP4FLG ;CHECK FOR OCCURENCE OF TRAP.
2564 017726 001401 SEC ;INDICATE SUCCESS.
2565 017730 000241 BEQ 60$ ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
2566 017732 004736 CLC ;INDICATE FAILURE.
2567 017734 000207 PASS 60$: ;RESTORE GPRS.
2567 017734 000207 RTS PC JSR PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
```

```
2569 .SBTTL GLOBAL SUBROUTINE - CLNRST -
2570 ;*****
2571 ;* - CLEAN RESET OF THE DEVICE UNDER TEST -
2572 ;* THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
2573 ;* THE DUT'S SELF-TEST IS SKIPPED, AND THE FIFO IS PURGED OF ANY ERROR
2574 ;* CODES, ETC.
2575 ;* IF THE RESET DOES NOT SUCCESSFULLY COMPLETE, THEN THE CARRY BIT IS
2576 ;* PASSED BACK TO THE CALLING ROUTINE (CLEAR).
2577 ;*
2578 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR
2579 ;* TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
2580 ;* ERRNBR - ERROR NUMBER FOR POSSIBLE ERROR REPORT.
2581 ;* ERRIBL - ERRTP,ERNBR,AND ERRMSG SET UP CORRECTLY.
2582 ;*
2583 ;* OUTPUTS: THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
2584 ;* CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
2585 ;* ERRBLK - VALUE MAY BE DESTROYED.
2586 ;* IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
2587 ;* TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
2588 ;*
2589 ;* CALLING SEQUENCE: JSR PC,CLNRST
2590 ;*
2591 ;* COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS ERRNBR.
2592 ;* THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
2593 ;*
2594 ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET,PUFIFO,RESETT.
2595 ;*****
2596
2597 017736 CLNRST:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
017736 004567 164030 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2598
2599 ;* RESET THE DUT.
2600 ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS FROM ERRNBR THRU ERRNBR+2.
2601 ; -
2602 017742 004767 001344 JSR PC,RESETT ;RESET THE DUT TO A KNOWN STATE.
2603 017746 103002 BCC 60$ ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
2604
2605 ;* PURGE THE FIFO OF ERROR CODES. SAVE ANY BMP CODES FOUND.
2606 ; -
2607 017750 004767 000522 JSR PC,PUFIFO ;PURGE THE FIFO.
2608
2609 017754 60$: ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
2610 017754 PASS ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
017754 004736 JSR PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
2611
2612 017756 000207 RTS PC ;CARRY BIT:IF CLEAR, THEN ABORT THE TEST.
```

```

2614 .SBTTL GLOBAL SUBROUTINE - CLR16W -
2615 ;** *****
2616 ;* - CLEAR SIXTEEN WORDS ROUTINE -
2617 ;* THIS SUBROUTINE CLEARS 16 WORDS STARTING WITH THE SPECIFIED WORD.
2618 ;*
2619 ;* INPUTS: R0 - ADDRESS OF THE FIRST WORD TO CLEAR.
2620 ;*
2621 ;* OUTPUTS: (R0) TO (R0+15) - 16 WORDS OF MEMORY ARE CLEARED TO 0.
2622 ;*
2623 ;* CALLING SEQUENCE: JSR PC,CLR16W
2624 ;*
2625 ;* COMMENTS:
2626 ;*
2627 ;* SUBORDINATE ROUTINES CALLED: NONE.
2628 ;** *****
2629
2630 CLR16W:: SAVE
2631 017760 004567 164006 JSR R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
2632 017764 012701 000020 ;CALL REGISTER SAVE SUBRT.
2633 017770 005020 ;SET THE LOOP COUNTER TO 16.
2634 017772 005301 ;CLEAR A WORD OF MEMORY.
2635 017774 001375 ;COUNT THIS LOOP.
2636 017776 004736 ;LOOP IF NOT 16 WORD CLEARED.
020000 000207 ;RESTORE GPRS.
PC,0(SP) ;RETURN TO PREG05 SUBRT.
RTS PC JSR

```

```

2638 .SBTTL GLOBAL SUBROUTINE - CNTERR -
2639 ;* *****
2640 ;* - COUNT ERROR ROUTINE -
2641 ;* THIS SUBROUTINE IS USED TO COUNT A "DATA" ERROR ON THE SPECIFIED
2642 ;* LINE. IT CHECKS WHETHER ERROR SUMMARY REPORTING IS ACTIVE, OR SHOULD
2643 ;* BE MADE ACTIVE ON THIS LINE, AND ACTIVATES IT IF NECESSARY.
2644 ;*
2645 ;* INPUTS: R5 - LINE NUMBER OF LINE UNDER CONSIDERATION.
2646 ;* ERCNTB - LABEL AT BASE OF ERROR COUNTERS TABLE.
2647 ;* ERSRPF - ERROR SUMMARY FLAGS (BIT SET IF LINE IN SUMMARY MODE).
2648 ;* NDERPT - NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE.
2649 ;*
2650 ;* OUTPUTS: CARRY - SET IF LINE IS IN ERROR SUMMARY MODE.
2651 ;* ERCNT - ERROR COUNTER INCREMENTED FOR SPECIFIED LINE.
2652 ;* ERSRPF - BIT SET IF LINE SHOULD BE IN SUMMARY MODE.
2653 ;*
2654 ;* CALLING SEQUENCE: JSR PC,CNTERR
2655 ;*
2656 ;* COMMENTS:
2657 ;*
2658 ;* SUBORDINATE ROUTINES CALLED: NONE.
2659 ;* - *****
2660
2661 020002 CNTERR:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
2662 020002 004567 163764 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2663 ;*
2664 ;* COUNT THE ERROR ON THE COUNTER FOR THE SPECIFIED LINE.
2665 ;*
2666 020006 006305 ASL R5 ;FORM WORD OFFSET FROM LINE NUMBER.
2667 020010 016501 002460 MOV ERCNTB(R5),R1 ;GET THE PRESENT ERROR COUNT FOR THIS LINE.
2668 020014 005201 INC R1 ;COUNT ERROR.
2669 020016 103402 BCS 2: ;OVERFLOW? YES, DON'T UPDATE COUNTER IN TABLE.
2670 020020 010165 002460 MOV R1,ERCNTB(R5) ;UPDATE ERROR COUNTER TABLE ENTRY.
2671 020024 005767 162172 2: TST NDERPT
2672 020030 001411 BEQ 60: ;SUMMARYS DISABLED? YES, EXIT WITH CARRY 0.
2673 020032 020167 162164 CMP R1,NDERPT ;NO, CHECK FOR ENOUGH ERRORS FOR SUMMARY USE.
2674 020036 101002 BHI 4: ;ENOUGH ERRORS TO USE SUMMARY? YES, GO HANDLE.
2675 020040 000241 CLC ;INDICATE NOT TO USE SUMMARY REPORT YET.
2676 020042 000404 BR 60: ;EXIT WITH CARRY 0.
2677 020044 056567 002404 4: BIS BITTBL(R5),ERSRPF ;SET THE ERROR SUMMARY FLAG FOR I.LINE.
2678 020052 000261 SEC ;INDICATE TO USE SUMMARY REPORT.
2679 020054 004736 60: PASS ;RESTORE GPRS.
2679 020056 000207 RTS PC JSR PC,B(SP) ;RETURN TO PREG05 SUBRT.

```

```
2681 .SBTTL GLOBAL SUBROUTINE - DELAY -
2682 ;*****
2683 ;* - DELAY SUBROUTINE -
2684 ;* THIS SUBROUTINE IS USED TO DELAY A VARIABLE NUMBER OF MILLI-SECONDS.
2685 ;*
2686 ;* INPUTS: R4 - CONTAINS THE NUMBER OF MS TO DELAY.
2687 ;* MSLCNT.
2688 ;*
2689 ;* OUTPUTS: NONE.
2690 ;*
2691 ;* CALLING SEQUENCE: JSR PC,DELAY
2692 ;*
2693 ;* COMMENTS: IF NO HARDWARE CLOCK INTERRUPTS ARE OCCURING, CONTROL-CS WILL
2694 ;* NOT BE HONORED FOR THE DURATION OF THE DELAY.
2695 ;*
2696 ;* SUBORDINATE ROUTINES CALLED: NONE.
2697 ;*****
2698
2699 020060 DELAY:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
020060 004567 163706 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2700 020064 010401 ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
2701 020066 012702 177777 ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
2702 020072 005003 ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
2703 020074 012704 020116 ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
2704 020100 004767 000130 ;DELAY THE REQUESTED # OF MS.
2705 020104 103002 ;EXIT ROUTINE IF WE TIMED-OUT.]
2706 020106 004767 000136 ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
2707 020112 ;RESTORE GPRS.
020112 004736 ;PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
2708 020114 000207
2709
2710 020116 177777
604: PASS JSR
MOV R4,R1 JSR
MOV #-1,R2
CLR R3
MOV #624,R4
JSR PC,MSLOOP
BCC 604
JSR PC,OOFS
624: .WORD -1 ;DUMMY, NON-ZERO WORD.
```

```

2712 .SBTTL GLOBAL SUBROUTINE - MSLGET -
2713 *****
2714 ;* - MILLI SECONDS LOOP WHICH RETURNS READ WORD AND REMAINING TIME -
2715 ;* THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
2716 ;* TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD. THE
2717 ;* CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
2718 ;* DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
2719 ;* THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
2720 ;* ROUTINE AND THEN ONCE EACH MILLI-SECOND THERE AFTER.
2721 ;* UPON RETURN, THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION
2722 ;* IS RETURNED BY THIS SUBROUTINE.
2723 ;*
2724 ;* INPUTS: R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
2725 ;* R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
2726 ;* R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
2727 ;* R4 - ADDRESS OF THE WORD TO TEST.
2728 ;* MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
2729 ;*
2730 ;* OUTPUTS: R0 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
2731 ;* R1 - REMAINING NUMBER OF MS IN TIME-OUT TIME.
2732 ;* CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
2733 ;*
2734 ;* CALLING SEQUENCE: JSR PC,MSLGET
2735 ;*
2736 ;* COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
2737 ;* CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
2738 ;* ON THE SYSTEM.
2739 ;* THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
2740 ;* DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
2741 ;* LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
2742 ;* IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
2743 ;* THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
2744 ;* IF THE CONDITION IS MET, FAILURE OTHERWISE.
2745 ;*
2746 ;*
2747 ;* SUBORDINATE ROUTINES CALLED: NONE.
2748 ;*
2749 ;* *****
2750 MSLGET:: SAVE JSR ,SAVE CONTENTS OF GPRS R0 THRU R5.
2751 020120 004567 163646 JSR R5,PREG05 ,CALL REGISTER SAVE SUBRT.
2752 ;*
2753 ;* SET UP MASK FOR REMOVING UNUSED BITS IN THE TEST WORD, AND CLEAR UNUSED
2754 ;* BITS IN THE DESIRED STATE WORD TO ALLOW DIRECT COMPARISON.
2755 ;*
2756 020124 005102 COM R2 ,GET MASK OF UNUSED BITS.
2757 020126 040203 BIC R2,R3 ,MASK OUT UNUSED BITS IN DESIRED STATE WORD.
2758 ;*
2759 ;* HANDLE THE TEST AND EXIT IF WE HAVE A 0 TIME-OUT VALUE.
2760 ;*
2761 020130 005701 TST R1 ,TEST THE TIME-OUT VALUE FOR ZERO.
2762 020132 001011 BNE 2# ,IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
2763 020134 011400 MOV (R4),R0 ,GET THE WORD TO TEST BEFORE EXITING.
2764 020136 010067 000070 MOV R0,62# ,SAVE VALUE SO WE CAN RETURN IT.
2765 020142 040200 BIC R2,R0 ,MASK OUT UNTESTED BITS OF WORD.
2766 020144 020003 CMP R0,R3 ,COMPARE AGAINST DESIRED STATE WORD.
2767 020146 000261 SEC ,INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
2767 020150 001420 BEQ 6# ,EXIT WITH SUCCESS IF WORDS ARE EQUAL.

```

```

2768 020152 000241          CLC          ;INDICATE FAILURE (TIME-OUT).
2769 020154 000416          BR          6$ ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
2770                                ;*
2771                                ; NON-ZERO TIME-OUT VALUE. LOOP, WAITING FOR CONDITION OR TIME-OUT.
2772                                ; -
2773 020156 016705 162170 2$: MOV MSLCNT,R5 ;LOAD MS LOOP COUNT.
2774 020162 011400 4$: MOV (R4),R0 ;GET THE WORD TO TEST.
2775 020164 010067 000042 MOV R0,62$ ;SAVE WORD IN CASE THIS IS THE LAST.
2776 020170 040200 BIC R2,R0 ;MASK OUT UNTESTED BITS OF WORD.
2777 020172 020003 CMP R0,R3 ;COMPARE AGAINST DESIRED STATE WORD.
2778 020174 000261 SEC ;SET CARRY IN CASE OF SUCCESS.
2779 020176 001405 BEQ 6$ ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
2780 020200 005305 DEC R5 ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2781 020202 001367 BNE 4$ ;LOOP IF MS NOT UP.
2782 020204 005301 DEC R1 ;DECREMENT THE MS TIME COUNT.
2783 020206 001363 BNE 2$ ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
2784 020210 000241 CLC ;CLEAR CARRY, WE TIMED-OUT.
2785                                ;*
2786                                ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME-OUT VALUE).
2787                                ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD. EXIT ROUTINE.
2788                                ; -
2789 020212 016700 000014 6$: MOV 62$,R0 ;PASS OUT THE LAST READ WORD.
2790 020216 010066 000002 60$: PASS R0,R1 ;RESTORE GPRS, EXCEPT THE FOLLOWING:
                                ;RO,ROSLOT(SP) ;PUT RO IN STACK SLOT.
                                ;R1,R1SLOT(SP) ;PUT R1 IN STACK SLOT.
                                ;PC,8(SP)+ ;RETURN TO PREG05 SUBRT.
                                ;R0 - LAST READ WORD CHECKED FOR CONDITION.
                                ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
                                ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
2791                                ;*
2792                                ; LOCAL STORAGE.
2793 020230 000207          RTS PC
2794                                ; -
2795                                ;*
2796                                ;
2797 020232 000000 62$: .WORD 0 ;STORAGE FOR THE LAST READ WORD.

```

```

2799 .SBTTL GLOBAL SUBROUTINE - MSLOOP -
2800 ;*****
2801 ;* - TEST LOOP SUBROUTINE -
2802 ;* THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
2803 ;* TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD. THE
2804 ;* CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
2805 ;* DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
2806 ;* THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
2807 ;* ROUTINE AND THEN ONCE EACH MILLI-SECOND THEREAFTER.
2808 ;*
2809 ;* INPUTS: R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
2810 ;* R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
2811 ;* R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
2812 ;* R4 - ADDRESS OF THE WORD TO TEST.
2813 ;* MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
2814 ;*
2815 ;* OUTPUTS: CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
2816 ;*
2817 ;* CALLING SEQUENCE: JSR PC,MSLOOP
2818 ;*
2819 ;* COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
2820 ;* CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
2821 ;* ON THE SYSTEM.
2822 ;* THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
2823 ;* DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
2824 ;* LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
2825 ;* IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
2826 ;* THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
2827 ;* IF THE CONDITION IS MET, FAILURE OTHERWISE.
2828 ;*
2829 ;* SUBORDINATE ROUTINES CALLED: MSLGET.
2830 ;*****
2831
2832 020234 MSLOOP:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
2833 020234 004567 163532 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2834
2835 ;*
2836 ;* CALLING THE MSLGET ROUTINE FROM THE MSLOOP ROUTINE ISOLATES THE CALLER OF
2837 ;* MSLOOP FROM THE RETURNED TEST WORD AND REMAINING TIME OUT VALUES.
2838 020240 004767 177654 JSR PC,MSLGET ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.
2839
2840 020244 601: PASS JSR ;RESTORE GPRS,
2841 020246 004736 PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
2841 020246 000207 RTS PC ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.

```

```
2843 .SBTTL GLOBAL SUBROUTINE - OOPS -
2844 ;* *****
2845 ;* - PROGRAM ABORT SUBROUTINE -
2846 ;* THIS SUBROUTINE IS USED TO ABORT THE PROGRAM WHEN A FATAL ERROR IS
2847 ;* DETECTED IN THE PROGRAM OR THE HOST SYSTEM HARDWARE. AN ERROR MESSAGE
2848 ;* IS PRINTED GIVING SOME INFORMATION ABOUT THE NATURE OF THE ABORT.
2849 ;*
2850 ;* INPUTS: R1 - ERROR CODE GIVING REASON FOR ABORT.
2851 ;*
2852 ;* OUTPUTS: AN ERROR MESSAGE IS PRINTED.
2853 ;* A LIST OF RETURN PC VALUES FOR ALL SUBROUTINE CALLS IS PRINTED.
2854 ;*
2855 ;* CALLING SEQUENCE: JSR PC,OOPS
2856 ;*
2857 ;* COMMENTS:
2858 ;*
2859 ;* SUBORDINATE ROUTINES CALLED: NONE.
2860 ;* - *****
2861
2862 020250 OOPS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2863 020250 004567 163516 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2864 ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
2864 020254 ERRSF 101,EM0101
2864 020254 104454 TRAP C#ERSF
2864 020256 000145 .WORD 101
2864 020260 020314 .WORD EM0101
2864 020262 000000 .WORD 0
2865 ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
2866 020264 PRINTF #EM0102
2866 020264 012746 020400 MOV #EM0102,-(SP)
2866 020270 012746 000001 MOV #1,-(SP)
2866 020274 010600 MOV SP,R0
2866 020276 104417 TRAP C#PNTF
2866 020300 062706 000004 ADD #4,SP
2867 2#: BREAK ;LOOK FOR OPERATOR CONTROL-C INPUT.
2867 020304 104422 TRAP C#BRK
2868 020306 000776
2869 60#: BR 2# ;INFINITE LOOP.
2869 020310 PASS ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
2870 020312 004736 JSR PC,0(SP). ;RETURN TO PREG05 SUBRT.
2871 020312 000207 ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.
2871
2872 020314 110 117 123 EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
2872 020317 124 040 103
2872 020322 117 115 120
2872 020325 125 124 105
2872 020330 122 040 110
2872 020333 101 122 104
2872 020336 127 101 122
2872 020341 105 040 117
2872 020344 122 040 123
2872 020347 117 106 124
2872 020352 127 101 122
2872 020355 105 040 102
2872 020360 125 107 040
2872 020363 105 116 103
2872 020366 117 125 116
2872 020371 124 105 122
```

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

.EVEN

```
2876 .SBTTL GLOBAL SUBROUTINE - PUFIFO -
2877 ;*****
2878 ;* - PURGE THE FIFO
2879 ;* THIS ROUTINE TRIES TO REMOVE ALL THE CHARACTERS FROM THE FIFO.
2880 ;* ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE BMP CODE QUEUE.
2881 ;*
2882 ;* INPUTS: RBUFA- CONTAINS THE ADDRESS OF THE RECEIVER.
2883 ;*
2884 ;*
2885 ;* OUTPUTS: CARRY BIT - INDICATES THE STATE OF THE FIFO, SET:= PURGED.
2886 ;* BMPCQ - THE CONTENTS OF THE BMP CODE QUEUE MAY BE UPDATED.
2887 ;*
2888 ;* CALLING SEQUENCE: JSR PC,PUFIFO
2889 ;*
2890 ;* COMMENTS:
2891 ;*
2892 ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
2893 ;*****
2894
2895 020476 PUFIFO::SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
020476 004567 163270 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2896 020502 012701 001000 MOV #512.,R1 ;SET MAXIMUM TRY COUNT OF 512.
2897 020506 016704 161526 MOV RBUFA,R4 ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
2898
2899 020512 011402 28: MOV (R4),R2 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
2900 020514 100016 BPL 68 ;EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.
2901
2902 ;*
2903 ;* CHECK IF THE READ CHARACTER IS ACTUALLY A BMP CODE.
2904 ;* IF IT IS, THEN SAVE IT ON THE BMP CODE QUEUE TO BE REPORTED LATER.
2905 020516 012700 070000 ;*
020522 040200 MOV #70000,R0 ;GENERATE A BIT MAP OF CHAR ERROR BITS
2907 020524 001006 BIC R2,R0 ;WHICH ARE NOT SET FOR CHAR.
2908 BNE 48 ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
2909
2910 ;*
2911 ;* CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
2912 020526 012700 000301 ;*
020532 040200 MOV #301,R0 ;CHECK IF BMP.
2913 020534 001002 BIC R2,R0 ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
2914 020536 004767 001470 BNE 48 ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
2915 JSR PC,SAVBMP ;SAVE BMP CODE ON THE QUEUE.
2916 020542 005301 48: DEC R1 ;DECREMENT THE TRY COUNT.
2917 020544 001362 BNE 28 ;LOOP TO TRY AGAIN.
2918 020546 000241 CLC ;CLEAR CARRY, TO INDICATE FIFO NOT PURGED.
2919 020550 000401 BR 608 ;EXIT WITH CARRY CLEAR.
2920 020552 000261 68: SEC ;SET CARRY, TO INDICATE FIFO PURGED.
2921
2922 020554 608: PASS ;RESTORE GPRS.
020554 004736 JSR PC,B(SP), ;RETURN TO PREG05 SUBRT.
2923 ;CARRY BIT, SET INDICATES FIFO PURGED.
2924 020556 000207 RTS PC
```

2926 .SBTTL GLOBAL SUBROUTINE - RDPDR -
2927 ;*****
2928 ;* - READ AND VERIFY DATA PATTERN FROM DEVICE REGISTERS ROUTINE -
2929 ;* THIS ROUTINE READS AND VERIFIES THE ROTATED DATA PATTERN WHICH HAS
2930 ;* BEEN WRITTEN BY THE WOPDR SUBROUTINE.
2931 ;* EACH ACTIVE LINE'S REGISTER'S CONTENTS IS READ AND COMPARED WITH THE
2932 ;* WRITTEN DATA.
2933 ;* AFTER THE UNUSED AND READ ONLY (RO) BITS ARE MASKED OUT, ANY ERRORS ARE
2934 ;* REPORTED FROM THIS ROUTINE.
2935 ;* THIS ROUTINE WILL TAKE INTO ACCOUNT THE TYPE OF WRITE OPERATION WHICH
2936 ;* WAS PERFORMED BY THE WOPDR SUBROUTINE.
2937 ;*
2938 ;* INPUTS: R2 - USED TO PASS IN THE DATA PATTERN TO BE ROTATED & VERIFIED.
2939 ;* R3 - BYTE INDICATOR (- => LO BYTE, + => HI BYTE, 0 => BOTH).
2940 ;* R4 - OPERATION TYPE INDICATOR (- => BIC, + => BIS, 0 => MOV).
2941 ;* ACTLNS - BIT MAP OF ACTIVE LINES ON THE DEVICE UNDER TEST.
2942 ;* CSRA - CONTAINS THE CSR ADDRESS OF THE DEVICE UNDER TEST.
2943 ;* DRADRT - BASE ADDRESS OF DEVICE REGISTER ADDRESS TABLE.
2944 ;* ERCNTB - LABEL AT BASE OF ERROR COUNTERS TABLE FOR LINES.
2945 ;* ERRMSG - SET UP WITH THE PROPER ERROR MESSAGE FOR THIS TEST.
2946 ;* ERRNBR - SET UP WITH THE PROPER ERROR NUMBER.
2947 ;* LPRO - EQUATED TO LPR REG OFFSET FROM DEVICE CSR ADDRESS.
2948 ;* NUMLNS - NUMBER OF LINES ON THE DEVICE UNDER TEST.
2949 ;* NDERPT - NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE.
2950 ;* TXBFCO - EQUATED TO TBUFFT REG OFFSET FROM DEVICE CSR ADDRESS.
2951 ;* UNBTB - BASE ADDRESS OF THE UNUSED BIT TABLE.
2952 ;*
2953 ;* OUTPUTS: ERROR MESSAGES MAY BE PRINTED AT THE OPERATOR'S CONSOLE.
2954 ;* ERCNT - ERROR COUNTERS TABLE IS UPDATED FOR LINE UNDER TEST.
2955 ;* ERRBLK - CONTENTS DESTROYED.
2956 ;* ERSRPF - ERROR SUMMARY FLAGS BIT SET IF LINE IN SUMMARY MODE.
2957 ;* UUT CSR - ALL BITS CLEARED, EXCEPT IND.ADR.REG FIELD DESTROYED.
2958 ;*
2959 ;* CALLING SEQUENCE: JSR PC,RDPDR
2960 ;*
2961 ;* COMMENTS: FOR BYTE ACCESSES, ONLY THE SPECIFIED BYTE IS VERIFIED.
2962 ;*
2963 ;* SUBORDINATE ROUTINES CALLED: ER1601,ROLDAP.
2964 ;* - *****
2965
2966 RDPDR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2967 020560 004567 163206 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2968 020564 012767 016352 163176 MOV #ER1601,ERRBLK ;SET UP THE ADDRESS OF THE ERROR REPORT RTN.
2969 ;*
2970 ; DETERMINE WHETHER REGISTER DATA SHOULD BE INVERTED FROM DATA PATTERN.
2971 ; -
2972 020572 005704 TST R4 ;CHECK THE OPERAND TYPE INDICATOR.
2973 020574 100001 BPL 21 ;BIC WRITE PERFORMED? NO, USE STANDARD DATA.
2974 020576 005102 COM R2 ;YES, INVERT THE DATA PATTERN.
2975 ;*
2976 ; SET UP OUTER LOOP.
2977 020600 005005 21: CLR R5 ;CLEAR LINE COUNTER TO SELECT LINE 0.
2978 ;*
2979 ; THE OUTER LOOP FOLLOWS. EACH PASS THROUGH THIS LOOP READS AND COMPARES DATA
2980 ; FROM ALL OF THE DEVICE REGISTERS FOR A PARTICULAR LINE IF THE LINE IS ACTIVE.
2981 ; -

```

2982 020602 010267 000222      41:  MOV    R2,701      ;SAVE THE OUTER LOOP DATA PATTERN.
2983 020606 010577 161424      MOV    R5,8CSRA    ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
2984 020612 010500              MOV    R5,R0
2985 020614 006300              ASL     R0
2986 020616 036067 002404 161400 BIT    BITTBL(R0),ACTLNS
2987 020624 001467              BEQ     161      ;IS THE LINE ACTIVE? NO, SKIP THE LINE.
2988 020626 012703 000004      MOV    4LPRO,R3    ;YES, INITIALIZE REGISTER OFFSET FOR LPR.
2989
2990      ;
2991      ; THE INNER LOOP FOLLOWS. EACH PASS THROUGH THIS LOOP READS AND COMPARES
2992      ; DATA FROM A DEVICE REGISTER.
2993 020632 010204      61:  MOV    R2,R4      ;SAVE THE INNER LOOP DATA PATTERN.
2994 020634 046302 002364      BIC     UNBITB(R3),R2 ;REMOVE UNUSED BITS FROM EXPECTED DATA.
2995 020640 016300 002236      MOV    DRADRT(R3),R0
2996 020644 005766 000010      TST     R3SLOT(SP) ;CHECK THE ACCESS TYPE INDICATOR.
2997 020650 001002      BNE     81      ;BYTE ACCESS? YES, GO PERFORM BYTE READ.
2998 020652 011001      MOV    (R0),R1 ;NO, PERFORM WORD READ OF DEVICE REGISTER.
2999 020654 000416      BR      121
3000 020656 100410      81:  BMI     101      ;LOW BYTE ACCESS? YES, GO DO LOW BYTE READ.
3001 020660 005200      INC     R0      ;HIGH BYTE ACCESS. FORM HIGH BYTE ADDRESS.
3002 020662 111001      MOVB   (R0),R1 ;READ THE HI BYTE OF THE DUT REGISTER.
3003 020664 000301      SWAB   R1      ;PUT HI BYTE BACK INTO THE HI BYTE.
3004 020666 042701 000377      BIC     4377,R1    ;REMOVE THE UNUSED BYTE IN ACTUAL DATA.
3005 020672 042702 000377      BIC     4377,R2    ;REMOVE THE UNUSED BYTE IN EXPECTED DATA.
3006 020676 000405      BR      121
3007 020700 111001      101:  MOVB   (R0),R1 ;READ THE LOW BYTE OF THE DUT REGISTER.
3008 020702 042701 177400      BIC     4177400,R1 ;REMOVE THE UNUSED BYTE.
3009 020706 042702 177400      BIC     4177400,R2 ;FORM EXPECTED LOW BYTE FOR COMAPARISON.
3010
3011 020712 046301 002364      121:  BIC     UNBITB(R3),R1 ;REMOVE UNUSED BITS FROM ACTUAL DATA.
3012 020716 020102      CMP     R1,R2    ;COMPARE ACTUAL AND EXPECTED DATA.
3013 020720 001414      BEQ     141      ;ACTUAL = EXPECTED? YES, SKIP ERROR.
3014 020722 004767 177054      JSR     PC,CTERR   ;NO, COUNT THE ERROR, CHECK FOR ERROR SUMMARY.
3015 020726 103411      BCS     141      ;USE ERROR SUMMARY? YES, SKIP ERROR.
3016      ;NO, REPORT "BAD BIT(S) IN DEVICE XXXXX REGISTER FOR LINE NN (D)."
3017 020730      ERROR
3018      ;
3019      ; TRAP C1ERROR
3020      ; EXIT THIS ROUTINE AND SET THE "EXIT ON ERROR" FLAG, IF EXTENDED ERROR
3021      ; REPORTING HAS NOT BEEN REQUESTED.
3022 020732 032767 000100 161260      BIT     4BIT06,OPTION ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3023 020740 001004      BNE     141      ;BRANCH IF IT HAS.
3024 020742 012767 000001 161330      MOV    41,EXDERR ;SET THE EXIT ON ERROR FLAG.
3025 020750 000425      BR      601      ;EXIT THE ROUTINE.
3026
3027 020752 010402      141:  MOV    R4,R2      ;RESTORE THE INNER LOOP DATA PATTERN.
3028 020754 004767 000444      JSR     PC,ROLDAP  ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
3029 020760 062703 000002      ADD     42,R3      ;SET REGISTER OFFSET TO THE NEXT REGISTER.
3030 020764 020327 000006      CMP     R3,4FSLSO ;CHECK THAT THIS IS NOT THE FIFOSIZE/DATA REG.
3031 020770 001002      BNE     151      ;AVOID ALTERING THE OFFSET IF IT ISN'T
3032 020772 062703 000002      ADD     42,R3      ;POINT AT THE NEXT REGISTER.
3033 020776 020327 000016      151:  CMP     R3,4TXBFC0 ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
3034 021002 003713      BLE     61      ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
3035
3036      ; BACK INTO THE OUTER LOOP. NOW SET UP FOR NEXT LINE. LOOP IF NOT DONE.
3037      ;

```

3038	021004	016702	000020	16:	MOV	70,R2			;SET UP TO ROTATE THE DATA PATTERN.
3039	021010	004767	000410		JSR	PC,ROLDAP			;ROTATE THE DATA PATTERN.
3040	021014	005205			INC	R5			;COUNT THIS LINE
3041	021016	020527	000020		CMP	R5,#NUMLNS			;COMPARE LINE COUNT WITH NUMBER OF LINES.
3042	021022	002667			BLT	4:			;LOOP IF SOME LINES NOT DONE.
3043									
3044	021024			60:	PASS				;RESTORE GPRS.
	021024	004736					JSR		;RETURN TO PREG05 SUBRT.
3045	021026	000207			RTS	PC			
3046									
3047	021030	000000		70:	.WORD	0			;STORAGE FOR DATA PATTERN OUTSIDE INNER LOOP.

```
3049 .SBTTL GLOBAL SUBROUTINE - REGTST -
3050 ;* *****
3051 ;* - REGISTERS TEST SUBROUTINE -
3052 ;* SUBROUTINE TO TEST THE DEVICE UNDER TEST (DUT) REGISTERS. THE USED
3053 ;* BITS OF THE REGISTERS ARE EITHER ALL CLEARED OR ALL SET AND THEN THE
3054 ;* DATA PATTERN IS WRITTEN AND VERIFIED USING EITHER WORD OR BYTE
3055 ;* ACCESSES IN READ/WRITE OR READ/MODIFY/WRITE MODE.
3056 ;*
3057 ;* INPUTS: R3 - BYTE INDICATOR (- => LOW, + => HIGH, 0 => BOTH BYTES).
3058 ;* R4 - ACCESS MODE (-1 => SET THEN BIC, 1 => CLEAR THEN BIS,
3059 ;* (-2 => SET THEN MOV, +2 CLEAR THEN MOV).
3060 ;* ERRNBR - SET UP WITH INITIAL ERROR NUMBER.
3061 ;*
3062 ;* OUTPUTS: GPRS0 - GPR SAVE AREA 0 IS DESTROYED.
3063 ;* DEVICE UNDER TEST REGISTERS ARE WRITTEN.
3064 ;* ERROR MESSAGES MAY BE PRINTED AT THE OPERATORS CONSOLE.
3065 ;*
3066 ;* CALLING SEQUENCE: JSR PC,REGTST
3067 ;*
3068 ;* COMMENTS: THIS ROUTINE LOOP 16 TIMES WRITING THE SAME DATA PATTERN
3069 ;* ROTATED LEFT ONCE EACH ITERATION.
3070 ;* THIS ROUTINE CAN REPORT ERRORS INITIAL ERRNBR THRU INITIAL+2.
3071 ;*
3072 ;* SUBORDINATE ROUTINES CALLED: RDPDR,ROLDAP,SWAPO,MDPDR
3073 ;* - - - - -
3074
3075 021032 REGTST:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
021032 004567 162734 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3076
3077 ;* SET UP THE GPRS FOR THE WRITTING OF THE DATA PATTERN.
3078 ;*
3079 021036 012705 000020 MOV #16,,R5 ;SET UP LOOP COUNTER TO COUNT 16 ITERATIONS.
3080 021042 012702 167410 MOV #167410,R2 ;INITIALIZE THE DATA PATTERN.
3081 021046 032704 000001 BIT #BIT0,R4 ;TEST FOR R/W ACCESS.
3082 021052 001001 BNE 2# ;R/M/W ACCESS? YES, R4 IS ALL SET UP.
3083 021054 005004 CLR R4 ;NO, INDICATE R/W ACCESS.
3084 021056
3085 2# :
3086 ;* SET UP THE GPRS FOR THE CLEARING OR SETTING OF ALL THE USED BITS.
3087 ;*
3088 021056 010400 MOV R4,R0 ;PASS OPERATION TYPE INDICATOR AROUND SWAPO.
3089 021060 004767 001272 JSR PC,SWAPO ;GET ALTERNATE GPR SET IN R1 THRU R5.
3090 021064 016701 162674 MOV ERRNBR,R1 ;SAVE THE INITIAL ERROR NUMBER.
3091 021070 010004 MOV R0,R4
3092 021072 005404 NEG R4 ;SET UP OP TYPE FOR CLEARING OR SETTING.
3093 021074 005002 CLR R2 ;SET UP CLEAR WRITE PATTERN.
3094 021076 026627 000012 000002 CMP R4,SLOT(SP),#2 ;TEST FOR CLEAR THEN MOV TEST SEQUENCE.
3095 021104 001401 BEQ 4# ;CLEAR THEN MOV? YES, LEAVE WRITE PAT CLEAR.
3096 021106 005102 COM R2 ;NO, SET ALL BITS OF WRITE PATTERN.
3097 021110 005003 4# : CLR R3 ;INDICATE THAT WORD ACCESSES SHOULD BE USED.
3098 021112 005000 CLR R0 ;SET ALTERNATE BYTE EXPECTED DATA PAT TO CLEAR.
3099 021114 026627 000012 177776 CMP R4,SLOT(SP),#-2 ;TEST FOR SET THEN MOV TEST SEQUENCE.
3100 021122 001001 BNE 6# ;SET THEN MOV? YES, LEAVE ALT BYTE PAT CLEAR.
3101 021124 005100 COM R0 ;NO, SET ALT BYTE EXPECTED DATA PAT TO ALL 1'S.
3102 021126 004767 001224 6# : JSR PC,SWAPO ;RESTORE SWAPPED GPR VALUES TO R1 THRU R5.
3103 ;*
3104 ; START OF DATA PATTERN LOOP.
```

```

3105
3106 021132
3107
3108
3109
3110
3111 021132 004767 001220
3112 021136 004767 002062
3113 021142 010167 162616
3114 021146 004767 177406
3115
3116
3117
3118
3119 021152 005767 161122
3120 021156 001035
3121
3122 021160 004767 001172
3123
3124
3125
3126
3127 021164 004767 002034
3128 021170 005267 162570
3129 021174 004767 177360
3130
3131
3132
3133
3134 021200 005767 161074
3135 021204 001022
3136
3137 021206 005703
3138 021210 001414
3139
3140
3141
3142 021212 010201
3143 021214 010002
3144 021216 005403
3145 021220 005267 162540
3146 021224 004767 177330
3147
3148
3149
3150
3151 021230 005767 161044
3152 021234 001006
3153
3154 021236 005403
3155 021240 010102
3156
3157
3158
3159 021242 004767 000156
3160 021246 005305
3161 021250 003330

```

```

;
;
; SET OR CLEAR ALL THE USED BITS OF THE DEVICE REGISTERS FOR ALL LINES.
; VERIFY THAT ALL THE BITS WERE SET OR CLEARED CORRECTLY.
;
JSR PC,SWAPO ;GET ALTERNATE GPRS FOR SETTING INTIAL STATES.
JSR PC,WDPDR ;GO CLEAR ALL USED REGISTER BITS, ALL LINES.
MOV R1,ERRNBR ;SET UP ERROR NUMBER TO INITIAL ERRNBR.
JSR PC,RDPDR ;VERIFY ALL USED REGISTER BITS, ALL LINES.
;
;EXIT THIS ROUTINE IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING
;HAS BEEN REQUESTED. I.E. EXCERR IS NON-ZERO.
;
TST EXCERR ;HAS AN ERROR BEEN FOUND ?
BNE 604 ;EXIT THIS ROUTINE IF IT HAS.
;
JSR PC,SWAPO ;RESTORE MAIN GPRS CONTENTS.
;
; WRITE DATA PATTERNS, ALL LOWER BYTE USED BITS, ALL REGISTERS, ALL LINES.
; VERIFY THAT THE DATA PATTERN WAS WRITTEN CORRECTLY.
;
JSR PC,WDPDR ;WRITE DATA PATTERN TO DEVICE REGISTERS.
INC ERRNBR ;SET ERROR NUMBER TO INITIAL+1.
JSR PC,RDPDR ;VERIFY DATA PATTERN IN ALTERED BYTE(S).
;
;EXIT THIS ROUTINE IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING
;HAS BEEN REQUESTED.
;
TST EXCERR ;HAS AN ERROR BEEN FOUND ?
BNE 604 ;EXIT THIS ROUTINE IF IT HAS.
;
TST R3 ;CHECK THE BYTE INDICATOR.
BEQ 104 ;WORD ACCESS? YES, SKIP SECOND BYTE CHECK.
;
; CHECK THAT THE ALTERNATE (UNMODIFIED) BYTE IS CLEAR OR SET AS EXPECTED.
;
MOV R2,R1 ;SAVE THE DATA PATTERN.
MOV R0,R2 ;GET THE ALTERNATE BYTE EXPECTED DATA.
NEG R3 ;INDICATE THAT OTHER BYTE IS TO BE CHECKED.
INC ERRNBR ;SET ERROR NUMBER TO INITIAL+2.
JSR PC,RDPDR ;VERIFY DATA PATS IN OTHER BYTES OF REGISTERS.
;
;EXIT THIS ROUTINE IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING
;HAS BEEN REQUESTED.
;
TST EXCERR ;HAS AN ERROR BEEN FOUND ?
BNE 604 ;EXIT THIS ROUTINE IF IT HAS.
;
NEG R3 ;RESTORE BYTE INDICATOR.
MOV R1,R2 ;RESTORE DATA PATTERN.
;
; PREPARE THE NEXT DATA PATTERN AND LOOP IF NOT DONE.
;
104: JSR PC,ROLDAP ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
DEC R5 ;COUNT THIS ITERATION OF THE LOOP.
BGT 84 ;ALL PATTERNS DONE? NO, LOOP.

```

```
3162  
3163 021252 016767 161166 162504 601: MOV GPRS08,ERRNBR ;YES, RESTORE ERROR NUMBER AND EXIT.  
3164 021260 PASS ;GET THE ERROR NUMBR FROM GPR SWAP STORAGE.  
021260 004736 ;RESTORE GPRS.  
3165 021262 000207 RTS PC JSR PC,0(SP); ;RETURN TO PREG05 SUBRT.
```

```
3167 .SBTTL GLOBAL SUBROUTINE - REPSMR -
3168 ;* *****
3169 ;* - REPORT ERROR SUMMARY ROUTINE -
3170 ;* THIS SUBROUTINE REPORTS AN ERROR SUMMARY FOR THOSE LINES WHICH HAVE
3171 ;* EXCEEDED THE NUMBER OF INDIVIDUAL ERRORS TO REPORT FOR A SINGLE LINE
3172 ;* IN A SINGLE TEST. THIS PARAMETER CAN BE SPECIFIED BY THE OPERATOR IF
3173 ;* HE/SHE ANSWERS THE SOFTWARE PARAMETER QUESTIONS.
3174 ;*
3175 ;* INPUTS: ERCNTB - LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
3176 ;*          ERRMSG - ADDRESS OF PRIMARY ERROR MESSAGE.
3177 ;*          ERRNBR - ERROR NUMBER OF ERRORS IN THIS ROUTINE.
3178 ;*          ERSRFR - "REPORT ERROR SUMMARY FOR LINE" FLAGS.
3179 ;*
3180 ;* OUTPUTS: ERRBLK - ADDRESS OF ERROR REPORTING ROUTINE (DESTROYED).
3181 ;*          SUMMARY MESSAGES MAY BE PRINTED AT THE OPERATOR CONSOLE.
3182 ;*
3183 ;* CALLING SEQUENCE: JSR PC,REPSMR
3184 ;*
3185 ;* COMMENTS: IF NO LINES HAVE EXCEEDED THE MAXIMUM NUMBER OF INDIVIDUAL
3186 ;*          ERRORS TO REPORT, NO MESSAGES ARE PRINTED BY THIS ROUTINE.
3187 ;*          ERROR SUMMARIES IN THIS ROUTINE ARE REPORTED AS ERRORS.
3188 ;*          THE CONTENTS OF ERRBLK ARE DESTROYED.
3189 ;*
3190 ;* SUBORDINATE ROUTINES CALLED:
3191 ;* -- *****
3192
3193 021264 REPSMR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3194 021264 004567 162502 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3195 021270 005767 161162 TST ERSRFR ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
3196 021274 001404 BEQ 604 ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
3197
3198 ;* WE HAVE SOME ERROR SUMMARIES TO REPORT.
3199 021276 012767 016656 162464 MOV #ER9004,ERRBLK ;SELECT ERROR REPORTING ROUTINE.
3200
3201 ;* REPORT
3202 ;* "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
3203 ;*
3204 021304 ERROR
3205 021304 104460 TRAP C#ERROR
3206 021306 004736 604: PASS ;RESTORE GPRS.
3207 021310 000207 RTS PC JSR PC,8(SP), ;RETURN TO PREG05 SUBRT.
```

```

3209 .SBTTL GLOBAL SUBROUTINE - RESETT -
3210 *****
3211 ;* - RESET DEVICE UNDER TEST -
3212 ;* THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
3213 ;* IF RESET DOES NOT SUCCESSFULLY COMPLETE, IE. TIME-OUT OCCURS, THEN
3214 ;* AN ABORT TEST ERROR MESSAGE IS REPORTED.
3215 ;*
3216 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR
3217 ;* TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
3218 ;* ERRBL- ERRTP,ERNBR,AND ERRMSG SET UP CORRECTLY.
3219 ;*
3220 ;* OUTPUTS: THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
3221 ;* CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
3222 ;* ERRBLK - VALUE MAY BE DESTROYED.
3223 ;* IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
3224 ;* TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
3225 ;*
3226 ;* CALLING SEQUENCE: JSR PC,RESETT
3227 ;*
3228 ;* COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERNBR
3229 ;* THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERNBR.
3230 ;*
3231 ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET.
3232 *****
3233
3234 021312 RESETT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3235 021312 004567 162454 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3236 021316 012702 000040 MOV #BIT05,R2 ;SET BIT MASK OF MASTER RESET BIT.
3237 ;*
3238 ;* TEST THE STATE OF THE MASTER RESET BIT IN THE CSR.
3239 ;* IF MR IS SET THEN WAIT FOR SELF-TEST TO COMPLETE.
3240 ;* IF TIME-OUT OCCURS, REPORT THE ERROR AND PASS-OUT ABORT TEST INDICATOR.
3241 ;*
3241 021322 016704 160710 MOV CSRA,R4 ;GET THE ADDRESS OF THE DUT'S CSR.
3242 021326 030214 BIT R2,(R4) ;CHECK STATE OF MASTER RESET BIT.
3243 021330 001406 BEQ 2# ;DON'T DELAY IF MR IS ALREADY CLEAR.
3244 021332 005003 CLR R3 ;SET UP DESIRED STATE OF MASTER RESET BIT.
3245 021334 012701 011610 MOV #5000.,R1 ;PASS TIME-OUT VALUE OF 5 SECONDS.
3246 021340 004767 176554 JSR PC,MSLGET ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
3247 021344 103012 BCC 4# ;GO REPORT ERROR IF TIMEOUT OCCURRED.
3248 ;*
3249 ;*
3250 ;* SET MASTER RESET BIT IN CSR. CLEAR TX AND RX ENABLE BITS, ETC.
3251 ;* SKIP THE SELFTEST.
3252 ;* TIME-OUT OF 5 SECS. JUST IN CASE THE SELF-TEST EXECUTES.
3253 ;*
3254 021346 010277 160664 2# MOV R2,BCSRA ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
3255 021352 004767 000722 JSR PC,SKPSTS ;TRY TO SKIP THE SELFTEST.
3256 ;*
3257 ;* SET SELF-TEST TIME-OUT OF 5 SECONDS, AND WAIT FOR M.R TO CLEAR.
3258 ;* IF TIME-OUT OCCURS, THEN REPORT THE FATAL ERROR AND PASS-OUT THE ABORT
3259 ;* TEST INDICATOR.
3260 ;*
3261 021356 005003 CLR R3 ;SET UP DESIRED STATE OF MASTER RESET BIT.
3262 021360 012701 011610 MOV #5000.,R1 ;PASS TIME-OUT VALUE OF 5 SECONDS.
3263 021364 004767 176530 JSR PC,MSLGET ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
3264 021370 103410 BCS 6# ;SKIP ERROR REPORT IF MR CLEARED IN TIME.

```

```

3265
3266
3267
3268
3269 021372 012701 012171
3270 021376 012767 016462 162364
3271
3272
3273 021404
      021404 104460
3274 021406 000241
3275 021410 000403
3276
3277
3278
3279
3280 021412 005067 160666
3281 021416 000261
3282
3283 021420
      021420 004736
3284
3285 021422 000207
3286

; *
; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET, TEST ABORTED".
; INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
; -
4:  MOV    #EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
    MOV    #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
    ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
    ; "TEST ABORTED"
    ERROR
; >>>> ERROR <<<<
                                TRAP    C#ERROR
    CLC
    BR      60:            ;INDICATE TEST IS TO BE ABORTED.
                                ;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).
; -
6:  CLR    IESTAT          ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
    SEC
                                ;INDICATE SUCCESS, CONTINUE TEST.
60: PASS
                                ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
                                PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
                                ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
                                JSR
    RTS    PC

```

```

3288 .SBTTL GLOBAL SUBROUTINE - ROLDAP -
3289 ;*****
3290 ;* - ROTATE LEFT DATA PATTERN
3291 ;* THIS ROUTINE ROTATES THE PASSED INPUT DATA PATTERN LEFT,WITHOUT GOING
3292 ;* THROUGH THE CARRY.THE CARRY IS INITIALLY SET OR CLEARED DEPENDING
3293 ;* UPON THE STATE OF THE MSB OF THE DATA PATTERN,BEFORE A ROL INSTRUCTION
3294 ;* IS EXECUTED.
3295 ;*
3296 ;* INPUTS: R2 - CONTAINS THE DATA PATTERN TO BE ROTATED
3297 ;*
3298 ;* OUTPUTS: R2 - CONTAINS THE ROTATED DATA PATTERN
3299 ;*
3300 ;* CALLING SEQUENCE: JSR PC,ROLDAP
3301 ;*
3302 ;* COMMENTS:
3303 ;*
3304 ;* SUBORDINATE ROUTINES CALLED: NONE
3305 ;*****
3306
3307 021424 ROLDAP::SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
021424 004567 162342 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3308 021430 005702 TST R2 ;CHECK MSB, AND CLEAR CARRY.
3309 021432 100001 BPL 24 ;BRANCH IF CLEAR.
3310 021434 000261 SEC ;SET CARRY IF MSB SET
3311 021436 006102 24: ROL R2 ;ROTATE DATA PATTERN LEFT
3312 021440 010266 000006 604: PASS R2 ;RESTORE GPRS,EXCEPT
021440 010266 000006 MOV R2,R2.LOT(SP) ;PUT R2 IN STACK SLOT.
021444 004736 JSR PC,B(S-)+ ;RETURN TO PREG05 SUBRT.
3313 ;R2 - CONTAINS THE ROTATED DATA PATTERN
3314 021446 000207 RTS PC

```

```
3316 .SBTTL GLOBAL SUBROUTINE - RSTRPT -
3317 ;* *****
3318 ;* - REPORT ANY RESET ERRORS ROUTINE -
3319 ;* THIS ROUTINE DETERMINES IF ANY ERROR CODES ARE AMONG THE DIAGNOSTIC
3320 ;* CODES REPORTED PLACED IN THE DUT RECEIVED CHARACTER FIFO BY THE
3321 ;* SELF-TEST. IF ANY NON BMP ERROR CODES ARE FOUND, OR IF OTHER ERRORS
3322 ;* ARE ENCOUNTERED, APPROPRIATE ERRORS ARE REPORTED. ANY BMP CODES THAT
3323 ;* ARE FOUND, ARE PLACED ON THE BMP CODE QUEUE TO BE REPORTED LATER.
3324 ;* THIS ROUTINE ALSO PURGES THE DUT FIFO LOOKING FOR ANY CHARACTERS
3325 ;* OR MODEM STATUS CODES. IF ANY ARE FOUND, ERRORS ARE REPORTED.
3326 ;*
3327 ;* INPUTS: ERRMSG - ADDRESS OF THE PRIMARY ERROR MESSAGE.
3328 ;*          ERRNBR - ERROR NUMBER OF FIRST ERROR REPORTED BY THIS ROUTINE.
3329 ;*          NUMLNS - EQUATED TO THE NUMBER OF LINE ON THE DUT.
3330 ;*          RBUFA - CONTAINS ADDRESS OF THE DUT RECEIVER FIFO.
3331 ;*
3332 ;* OUTPUTS: CARRY - SUCCESS FLAG (SET IF FIFO CLEARED SUCCESSFULLY).
3333 ;*          ERRBLK - ADDRESS OF THE ERROR REPORT ROUTINE (DESTROYED).
3334 ;*          ERROR MESSAGES CAN BE PRINTED AT THE OPERATORS CONSOLE.
3335 ;*
3336 ;* CALLING SEQUENCE: JSR PC,RSTRPT
3337 ;*
3338 ;* COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERRNBR
3339 ;*            THRU INITIAL ERRNBR+4.
3340 ;*            THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
3341 ;*
3342 ;* SUBORDINATE ROUTINES CALLED: ER0503,ER9007,ER9008,SAVBMP.
3343 ;* -- *****
3344
3345 021450 RSTRPT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3346 021450 004567 162316 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3347
3348 ;* READ CORRECT NUMBER (NUMBER OF LINE ON DUT) OF CHARS FROM THE FIFO.
3349 ;* VERIFY THAT EACH CHAR IS A SELFTEST SUCCESS CODE.
3350 ;*
3351 021454 005003 ;CLR R3 ;CLEAR THE CODE COUNTER.
3352 021456 016705 162302 MOV ERRNBR,R5 ;SAVE ERRNBR FOR RESTORATION LATER.
3353 021462 017702 160552 MOV RBUFA,R2 ;READ A CHAR FROM THE DUT FIFO.
3354 021466 100422 BMI 4$ ;SKIP ERROR IF DATA.VALID SET FOR CHAR.
3355
3356 ;* WE EXPECT A SELFTEST CODE, BUT THIS FIFO SLOT IS EMPTY.
3357 021470 010567 162270 MOV R5,ERRNBR ;RESTORE ERROR NUMBER TO INITIAL VALUE.
3358 021474 012701 015123 MOV #EM9018,R1 ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
3359 021500 012767 016772 162262 MOV #ER9007,ERRBLK ;SELECT PROPER ERROR REPORT ROUTINE.
3360
3361 ;* REPORT ERROR WITH NUMBER INITIAL ERRNBR.
3362 ;* "NO SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET."
3363 ;*
3364 021506 104460 ERROR ;>>>> ERROR <<<<.
3365 021506 TRAP C#ERROR
3366
3367 ;* EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3368 021510 032767 000100 160502 BIT #BIT06,OPTION ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3369 021516 001003 BNE 3$ ;AVOID SET THE FLAG IF IT HAS.
3370 021520 012767 000001 160552 MOV #1,EXOERR ;SET THE EXIT ON ERROR FLAG
```

```
3371
3372
3373
3374 021526 000261
3375 021530 000167 000406
3376
3377
3378
3379 021534 012700 070001
3380 021540 040200
3381 021542 001042
3382
3383
3384
3385
3386 021544 032702 000200
3387 021550 001462
3388 021552 120227 000203
3389 021556 001457
3390 021560 120227 000201
3391 021564 001454
3392 021566 012700 000300
3393 021572 040200
3394 021574 001003
3395 021576 004767 000430
3396 021602 000445
3397
3398
3399
3400 021604 010567 162154
3401 021610 005267 162150
3402 021614 012701 015150
3403 021620 012767 017062 162142
3404
3405
3406
3407
3408 021626
3409 021626 104460
3410
3411
3412 021630 032767 000100 160362
3413 021636 001027
3414
3415 021640 012767 000001 160432
3416 021646 000534
3417
3418
3419
3420
3421 021650 010567 162110
3422 021654 062767 000002 162102
3423 021662 012701 015133
3424 021666 012767 016772 162074
3425
3426

;+
; INIDICATE "SUCCESS" (BECAUSE FIFO IS PURGED), AND EXIT THIS ROUTINE.
;-
3#: SEC ;SET SUCCESS FLAG.
JMP 60# ;EXIT ROUTINE.
;+
; DETERMINE IF THIS IS NOT A SELFTEST CODE.
;-
4#: MOV #70001,R0 ;GENERATE BIT MAP OF ANY CLEAR ERROR BITS OR
BIC R2,R0 ; BIT 0 WHICH ARE CLEAR.
BNE 8# ;GO TO REPORT ERROR IF THIS IS NOT A TEST CODE.
;+
; WE HAVE A TEST CODE (EITHER BMP OR SELFTEST CODE).
; DETERMINE WHAT TYPE OF CODE WE HAVE.
;-
BIT #BIT7,R2 ;TEST ROM VERSION CODE INDICATOR BIT.
BEQ 10# ;SKIP ERRORS IF SELFTEST ROM VERSION CODE.
CMPB R2,#203 ;CHECK IF SKIP SELF TEST CODE.
BEQ 10# ;SKIP ERROR REPORT IF SKIP SELF TEST CODE FOUND
CMPB R2,#201 ;CHECK IF NULL CODE PRESENT.
BEQ 10# ;SKIP ERROR REPORT IF SELF TEST NULL CODE.
MOV #300,R0 ;TEST CODE TYPE BITS FOR BOTH CODE
BIC R2,R0 ; TYPE BITS SET (INDICATING BMP CODE).
BNE 6# ;IF IT IS NOT A BMP CODE GO REPORT ERROR.
JSR PC,SAVBMP ;SAVE THE BMP CODE ON THE QUEUE.
BR 10# ;GO GET THE NEXT CHARACTER FROM THE FIFO.
;+
; WE HAVE A SELFTEST ERROR CODE.
;-
6#: MOV R5,ERRNBR ;RESTORE ERROR NUMBER TO INITIAL VALUE.
INC ERRNBR ;CALCULATE INITIAL ERROR NUMBER PLUS 1.
MOV #EM9020,R1 ;PASS ERROR MESSAGE INFO TO ER9008 ROUTINE.
MOV #ER9008,ERRBLK ;SELECT PROPER ERROR REPORT ROUTINE.
;+
; REPORT ERROR WITH NUMBER INITIAL ERRNBR + 1.
; "UNEXPECTED SELFTEST ERROR CODE FOR LINE NN IN FIFO AFTER RESET:"
;-
ERROR ; >>>> ERROR <<<<.
TRAP C#ERROR
;+
; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
;-
BIT #BIT06,OPTION ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
BNE 10# ;AVOID SET THE FLAG IF IT HAS AND GO TO
;THE END OF THE LOOP.
MOV #1,EXOERR ;SET THE "EXIT ON ERROR" FLAG
BR 50# ;EXIT THE ROUTINE WITH FAILURE SINCE THE FIFO
;IS NOT PRUGED.
;+
; WE HAVE A NON-SELFTEST CODE (EITHER BMP CODE OR DATA CHAR).
;-
8#: MOV R5,ERRNBR ;RESTORE ERROR NUMBER TO INITIAL VALUE.
ADD #2,ERRNBR ;CALCULATE INITIAL ERROR NUMBER PLUS 2.
MOV #EM9019,R1 ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
MOV #ER9007,ERRBLK ;SELECT PROPER ERROR REPORT ROUTINE.
;+
; REPORT ERROR WITH NUMBER INITIAL ERRNBR + 2.
```

```
3427      ; 'NON SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET.'
3428      ;
3429      021674      ; ERROR          >>>>> ERROR <<<<<,          TRAP      C!ERROR
          021674      104460
3430      ;
3431      ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3432      ;
3433      021676      032767      000100      160314      BIT      #BIT06,OPTION      ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3434      021704      001004      BNE      101          ;AVOID SET THE FLAG IF IT HAS AND GO TO
3435      ;THE END OF THE LOOP.
3436      021706      012767      000001      160364      MOV      #1,EXOERR      ;SET THE "EXIT ON ERROR" FLAG
3437      021714      000511      BR       501          ;EXIT THE ROUTINE WITH FAILURE.
3438
3439
3440      ;
3441      ; END OF LOOP, LOOP IF NOT ALL CHARS HAVE BEEN READ FROM THE FIFO.
3442      021716      005203      101:      INC      R3          ;SET CODE COUNTER FOR NEXT ITERATION OF LOOP.
3443      021720      020327      000010      CMP      R3,#8.      ;TEST FOR ALL CODES READ.
3444      021724      002656      BLT      21          ;LOOP IF NOT CHARS READ FROM FIFO.
3445
3446      ;
3447      ; PURGE THE FIFO UNTIL DATA.VALID IS CLEAR OR UNTIL TOO MANY CHARS ARE READ.
3448      021726      012704      000022      MOV      #18.,R4          ;INITIALIZE THE CHARACTER COUNTER.
3449      021732      010567      162026      MOV      R5,ERRNBR      ;GET INITIAL VALUE OF THE ERROR NUMBER.
3450      021736      062767      000003      162020      ADD      #3,ERRNBR      ;CALCULATE ERROR NUMBER OF NEXT ERROR.
3451      021744      012767      017062      162016      MOV      #ER9008,ERRBLK      ;SELECT PROPER ERROR REPORT ROUTINE.
3452      021752      017702      160262      121:      MOV      BRBUFA,R2      ;READ A CHARACTER FROM THE DUT FIFO.
3453      021756      000261      SEC          ;INDICATE SUCCESS IN CASE DATA.VALID IS CLEAR.
3454      021760      100070      BPL      601          ;EXIT ROUTINE WITH SUCCESS IF DATA.VALID CLEAR.
3455
3456      ;
3457      ; WE HAVE A CHARACTER.
3458      ; DETERMINE IF CHARACTER IS A DATA CHARACTER.
3459      021762      012700      070000      MOV      #70000,R0      ;TEST BITS 12 THRU 14 OF THE
3460      021766      040200      BIC      R2,R0      ; CODE READ FROM THE DUT FIFO.
3461      021770      001403      BEQ      141          ;SKIP THIS ERROR IF CODE IS NOT A DATA CHAR.
3462
3463      ;
3464      ; WE HAVE AN UNEXPECTED DATA CHARACTER: SET UP AND GO TO REPORT ERROR.
3465      021772      012701      015174      MOV      #EM9022,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3466      021776      000423      BR       221          ;GO TO REPORT THIS ERROR.
3467
3468      ;
3469      ; WE HAVE AN UNEXPECTED CODE.
3470      ; DETERMINE IF THE CODE IS A MODEM STATUS CODE.
3471      022000      032702      000001      141:      BIT      #BIT0,R2      ;TEST MODEM STATUS INDICATOR BIT OF CODE.
3472      022004      001003      BNE      161          ;SKIP THIS ERROR IF NOT MODEM STATUS CODE.
3473
3474      ;
3475      ; WE HAVE A MODEM STATUS CODE: SET UP AND GO TO REPORT ERROR.
3476      022006      012701      015213      MOV      #EM9023,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3477      022012      000415      BR       221          ;GO TO REPORT THIS ERROR.
3478
3479      ;
3480      ; WE HAVE AN ONBOARD TEST CODE.
3481      ; DETERMINE IF THIS CODE IS A BMP CODE.
3482      022014      032702      000200      161:      BIT      #BIT7,R2      ;TEST THE ROM VERSION BIT OF THE CODE.
```

```
3483 022020 001404      BEQ      181      ;GOTO SET UP FOR SELFTEST CODE IF ROM VERSION.
3484 022022 012700 000300      MOV      #300,R0
3485 022026 040200      BIC      R2,R0      ;TEST THE ERROR TYPE BITS OF THE CODE.
3486 022030 001403      BEQ      201      ;SKIP THIS ERROR IF BMP CODE.
3487
3488      ;*
3489      ; WE HAVE A SELFTEST CODE: SET UP AND GO TO REPORT ERROR.
3490 022032 012701 015235 181:      MOV      #EM9024,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3491 022036 000403      BR        221      ;GO TO REPORT THIS ERROR.
3492
3493      ;*
3494      ; WE HAVE A BMP CODE: SAVE IT ON THE QUEUE.
3495 022040 004767 000166 201:      JSR      PC,SAVBMP      ;SAVE THE BMP CODE ON THE QUEUE.
3496 022044 000411      BR        241
3497
3498      ;*
3499      ; REPORT THE ERROR WITH ERROR NUMBER OF INITIAL ERRNBR + 3.
3500      ; "UNEXPECTED XXX XXXX FOR LINE NN IN FIFO AFTER RESET:"
3501 022046 104460 221:      ERROR
3502      ;          >>>>> ERROR <<<<<.
3503      ;          TRAP      C1ERROR
3504      ;*
3505      ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3506 022050 032767 000100 160142      BIT      #BIT06,OPTION      ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3507 022056 001004      BNE      241      ;AVOID SETTING THE FLAG IF IT HAS AND GO TO
3508 022060 012767 000001 160212      MOV      #1,EXOERR      ;THE END OF THE LOOP.
3509 022066 000424      BR        501      ;SET THE "EXIT ON ERROR" FLAG
3510      ;EXIT THE ROUTINE WITH FAILURE.
3511
3512      ;*
3513      ; END OF LOOP.
3514      ; COUNT THE CHARACTER WE JUST RECEIVED, AND CHECK FOR TOO MANY RECEIVED.
3515 022070 005304 241:      DEC      R4      ;COUNT THIS CHARACTER.
3516 022072 001327      BNE      121      ;LOOP IF NOT TOO MANY CHARACTERS PURGED.
3517
3518      ;*
3519      ; WE READ TOO MANY VALID CHARACTERS WHILE TRYING TO PURGE THE FIFO.
3520      ; REPORT ERROR AND EXIT WITHOUT SUCCESS.
3521      ; "FIFO WILL NOT PURGE (DATA VALID STUCK SET), REMAINDER OF TEST SKIPPED."
3522 022074 012701 015012      MOV      #EM9017,R1      ;SELECT PROPER ERROR MESSAGE.
3523 022100 010567 161660      MOV      R5,ERRNBR      ;GET INITIAL ERROR NUMBER.
3524 022104 062767 000004 161652      ADD      #4,ERRNBR      ;CALCULATE INITIAL ERRNBR + 4.
3525 022112 012767 016072 161650      MOV      #ER0503,ERRBLK      ;SELECT PROPER ERROR REPORT ROUTINE.
3526      ;PRINT ERROR REPORT.
3527 022120 104460 221:      ERROR
3528      ;          >>>>> ERROR <<<<<.
3529      ;          TRAP      C1ERROR
3530      ;*
3531      ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3532 022122 032767 000100 160070      BIT      #BIT06,OPTION      ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3533 022130 001003      BNE      501      ;AVOID SETTING THE FLAG IF IT HAS.
3534 022132 012767 000001 160140      MOV      #1,EXOERR      ;SET THE "EXIT ON ERROR" FLAG
3535 022140 000241 501:      CLC      ;CLEAR THE SUCCESS FLAG.
3536
3537 022142 601:      PASS      ;RESTORE GPRS.
```

D8

CZDHUAO DMU 11 FUNC TST PART1 MACRO M1200 12 DEC-83 16:08 PAGE 67 4
GLOBAL SUBROUTINE - RSTRPT

SEQ 94

022142 004736
3538 022144 000207

RTS PC

JSR

PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
; CARRY - SUCCESS FLAG (SET IF FIFO IS PURGED).

```
3540 .SBTTL GLOBAL SUBROUTINE - RXIEO -
3541 ;* *****
3542 ;* - RECEIVER INTERRUPT DISABLE -
3543 ;* THIS ROUTINE IS USED TO DISABLE RECEIVER INTERRUPTS IN THE DHU11.
3544 ;*
3545 ;* INPUTS: NONE.
3546 ;*
3547 ;* OUTPUTS: THE RX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
3548 ;* IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3549 ;* ENABLE BITS.
3550 ;*
3551 ;* CALLING SEQUENCE: JSR PC,RXIEO
3552 ;*
3553 ;* COMMENTS: THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3554 ;* THE DUT CSR ARE DESTROYED.
3555 ;*
3556 ;* SUBORDINATE ROUTINES CALLED: NONE.
3557 ;* *****
3558 022146 010046 RXIEO:: MOV RO,-(SP) ;SAVE CONTENTS OF RO ON THE STACK.
3559 022150 104440 GETPRI -(SP) ;SAVE PROCESSOR PRIORITY ON STACK.
3560 022152 010046 TRAP C#GPRI
3560 022154 SETPRI #PRI07 ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
3561 022162 012700 000340 MOV RO,-(SP)
3562 022170 042767 137777 160114 TRAP C#PRI07,RO
3563 022176 016777 160110 160040 BIC #137777,IESTAT ;CLEAR RX.INT.ENBL BIT IN IESTAT.
3563 022176 012600 MOV IESTAT,BCSRA ;DISABLE RX INTERRUPTS.
3564 022202 012600 SETPRI (SP). ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
3565 022204 000207 TRAP C#SPRI
3565 MOV (SP)+,RO ;RESTORE RO.
3565 RTS PC
```

```

3567 .SBTTL GLOBAL SUBROUTINE - RXIE1 -
3568 ;* *****
3569 ;* - RECEIVER INTERRUPT ENABLE *****
3570 ;* THIS ROUTINE IS USED TO ENABLE RECEIVER INTERRUPTS IN THE DHU11.
3571 ;*
3572 ;* INPUTS: NONE.
3573 ;*
3574 ;* OUTPUTS: THE RX.INT.ENBL BIT IS SET IN THE DUT CSR.
3575 ;* IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3576 ;* ENABLE BITS.
3577 ;*
3578 ;* CALLING SEQUENCE: JSR PC,RXIE1
3579 ;*
3580 ;* COMMENTS: THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3581 ;* THE DUT CSR ARE DESTROYED.
3582 ;*
3583 ;* SUBORDINATE ROUTINES CALLED: NONE.
3584 ;* -- *****
3585
3586 022206 052767 000100 160070 RXIE1:: BIS #BIT06,IESTAT ;SET RX.INT.ENBL BIT IN IESTAT.
3587 022214 042767 137677 160062 BIC #137677,IESTAT ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
3588 022222 016777 160056 160006 MOV IESTAT,BCSRA ;ENABLE RX INTERRUPTS.
3589 022230 000207 RTS PC

```

```
3591 .SBTTL GLOBAL SUBROUTINE - SAVBMP -
3592 ;* *****
3593 ;* - SAVE BMP CODES ROUTINE -
3594 ;* THIS ROUTINE SAVES THE PARAMETER PASSED IN, ONTO THE BMP CODE QUEUE
3595 ;* TOGETHER WITH THE NUMBER OF THE CURRENTLY EXECUTING TEST.
3596 ;*
3597 ;* INPUTS: R2 - CONTAINS THE BMP CODE THAT IS TO BE PLACED ON THE QUEUE.
3598 ;* BMPCQP - CONTAINS ADDRESS OF NEXT LOCATION IN THE BMP QUEUE.
3599 ;* BMPCQB - LABEL AT BASE OF THE BMP CODE QUEUE.
3600 ;* BMPCQE - LABEL OF NEXT LOCATION AFTER THE END OF THE BMP QUEUE.
3601 ;* TSTNUM - CONTAINS THE NUMBER OF THE CURRENT TEST.
3602 ;*
3603 ;* OUTPUTS: BMPCQP - INCREMENTED BY 4.
3604 ;* THE CONTENTS OF THE BMP CODE QUEUE ARE UPDATED.
3605 ;*
3606 ;* CALLING SEQUENCE: JSR PC,SAVBMP
3607 ;*
3608 ;* COMMENTS: IF THE OVERFLOW OCCURS THEN THE LAST LOCATION WILL BE
3609 ;* OVERWRITTEN BY ANY SUBSEQUENT ATTEMPTS TO UPDATE THE QUEUE.
3610 ;*
3611 ;* SUBORDINATE ROUTINES CALLED: NONE.
3612 ;* *****
3613
3614 022232 SAVBMP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3615 022232 004567 161534 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3616 022236 016704 160256 MOV BMPCQP,R4 ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
3617 022242 116724 160052 MOV TSTNUM,(R4)+ ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
3618 022246 005204 INC R4 ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
3619 022250 042702 177400 BIC #177400,R2 ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
3620 022254 010224 MOV R2,(R4)+ ;SAVE THE BMP CODE ON THE QUEUE.
3621 022256 020427 002722 CMP R4,#BMPCQE ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
3622 022262 103402 BLO 2# ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
3623 022264 162704 000004 SUB #4,R4 ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
3624 022270 010467 160224 2#: MOV R4,BMPCQP ;SAVE THE POINTER.
3625 022274 60#: PASS ;RESTORE GPRS.
3626 022276 004736 JSR PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
3626 022276 000207 RTS PC
```

```
3628 .SBTTL GLOBAL SUBROUTINE - SKPSTS -
3629 ;** *****
3630 ;* - SKIP SELFTEST ROUTINE -
3631 ;* THIS SUBROUTINE IS USED TO SKIP THE SELFTEST AFTER A DUT RESET HAS BEEN
3632 ;* INITIATED. IT MUST BE ENTERED IMMEDIATELY AFTER SETTING THE DUT MASTER
3633 ;* RESET ROUTINE OR AFTER THE EXECUTION OF A BUS RESET (BECAUSE OF TIMING
3634 ;* CONSIDERATIONS).
3635 ;*
3636 ;* INPUTS: CSRA - CONTAINS ADDRESS OF THE DUT CSR.
3637 ;* TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
3638 ;*
3639 ;* OUTPUTS: SKIP SELFTEST CODES ARE WRITTEN TO THE DUT REGISTERS.
3640 ;*
3641 ;* CALLING SEQUENCE: JSR PC,SKPSTS
3642 ;*
3643 ;* COMMENTS:
3644 ;*
3645 ;* SUBORDINATE ROUTINES CALLED: DELAY.
3646 ;-- *****
3647
3648 022300 SKPSTS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3649 022300 004567 161466 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3650 022310 004767 175544 MOV #10.,R4 ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
3651 ; JSR PC,DELAY ;DELAY FOR 10 MILLI-SECONDS.
3652 ;*
3653 ; WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
3654 022314 012701 000060 ;--
3655 MOV #NUMLNS!BIT05,R1 ;FORM IND.ADR.REG FIELD (PLUS M.R. BIT) WORD.
3656 ;THE ABOVE INCLUSION OF THE M.R. BIT IS NECESSARY BECAUSE OF THE
3657 022320 012703 052525 ; LACK OF A M.R. BIT WRITE LOCK-OUT ON THE DHU-11.
3658 022324 005301 4: MOV #52525,R3 ;INITIALISE THE SKIP SELF-TEST CODE.
3659 022326 016704 157704 DEC R1 ;SELECT THE NEXT SET OF DEVICE REGISTERS.
3660 022332 010124 MOV CSRA,R4 ;GET THE ADDRESS OF THE CSR OF THE DUT.
3661 022334 010324 MOV R1,(R4)+ ;SELECT A BANK OF DUT REGISTERS.
3662 022336 020467 157712 MOV R3,(R4)+ ;WRITE THE CODE TO A DUT REGISTER.
3663 022342 103774 CMP R4,TXBFCA ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
3664 022344 032701 000017 BLO 6: ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
3665 022350 001365 BIT #17,R1 ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
3666 BNE 4: ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.
3667 022352 60: PASS ;RESTORE GPRS.
3668 022352 004736 JSR PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
3668 022354 000207 RTS PC
```

```

3670 .SBTTL GLOBAL SUBROUTINE - SWAPO -
3671 ;* *****
3672 ;* - SWAP GPRS WITH GPR SET 0 ROUTINE -
3673 ;* THIS SUBROUTINE SWAPS THE PRESENT CONTENTS OF GPRS R1 THRU R5 WITH
3674 ;* THE CONTENTS OF THE NUMBER ZERO GPR SAVE AREA. THE CONTENTS OF R0
3675 ;* ARE NOT ALTERED BY THIS SUBROUTINE.
3676 ;*
3677 ;* INPUTS: GPR CONTENTS R1 THRU R5.
3678 ;* GPRS08 - LABEL AT BASE OF GPR SAVE AREA NUMBER ZERO.
3679 ;*
3680 ;* OUTPUTS: R1 THRU R5 CONTAIN THE PREVIOUS CONTENTS OF GPR SAVE AREA
3681 ;* ZERO WORDS 1 THRU 5 RESPECTIVELY.
3682 ;* GPRS0 - GPR SAVE AREA 0 WORDS 1 THRU 5, CONTAIN PREVIOUS
3683 ;* CONTENTS OF GPRS R1 THRU R5 RESPECTIVELY.
3684 ;*
3685 ;* CALLING SEQUENCE: JSR PC,SWAPO
3686 ;*
3687 ;* COMMENTS: THE STATE OF THE CARRY FLAG IS NOT ALTERED BY THIS ROUTINE.
3688 ;*
3689 ;* SUBORDINATE ROUTINES CALLED: NONE.
3690 ;* - *****
3691
3692 022356 010046 SWAPO:: MOV R0,-(SP) ;SAVE THE CONTENTS OF R0.
3693 ;*
3694 ;* LOAD THE STACK FROM THE GPRS.
3695 ;*
3696 022360 010146 MOV R1,-(SP) ;SAVE THE CONTENTS OF R1.
3697 022362 010246 MOV R2,-(SP) ;SAVE THE CONTENTS OF R2.
3698 022364 010346 MOV R3,-(SP) ;SAVE THE CONTENTS OF R3.
3699 022366 010446 MOV R4,-(SP) ;SAVE THE CONTENTS OF R4.
3700 022370 010546 MOV R5,-(SP) ;SAVE THE CONTENTS OF R5.
3701 ;*
3702 ;* LOAD THE GPRS FROM THE GPR SAVE AREA 0.
3703 ;*
3704 022372 012700 002444 MOV #GPRS08,R0 ;GET THE BASE ADDRESS OF GPR SAVE AREA 0.
3705 022376 012001 MOV (R0)+,R1 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 1.
3706 022400 012002 MOV (R0)+,R2 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 2.
3707 022402 012003 MOV (R0)+,R3 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 3.
3708 022404 012004 MOV (R0)+,R4 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 4.
3709 022406 012005 MOV (R0)+,R5 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 5.
3710 ;*
3711 ;* LOAD THE GPR SAVE AREA 0 FROM THE STACK.
3712 ;*
3713 022410 012640 MOV (SP)+,-(R0) ;LOAD GPR SAVE AREA 0 WORD 5 WITH SAVED R5.
3714 022412 012640 MOV (SP)+,-(R0) ;LOAD GPR SAVE AREA 0 WORD 4 WITH SAVED R4.
3715 022414 012640 MOV (SP)+,-(R0) ;LOAD GPR SAVE AREA 0 WORD 3 WITH SAVED R3.
3716 022416 012640 MOV (SP)+,-(R0) ;LOAD GPR SAVE AREA 0 WORD 2 WITH SAVED R2.
3717 022420 012640 MOV (SP)+,-(R0) ;LOAD GPR SAVE AREA 0 WORD 1 WITH SAVED R1.
3718 ;*
3719 022422 012600 MOV (SP)+,R0 ;RESTORE THE INITIAL VALUE OF R0.
3720
3721 022424 000207 RTS PC

```

```

3723 .SBTTL GLOBAL SUBROUTINE - TSABRT -
3724 ;* *****
3725 ;* - TEST ABORT ROUTINE -
3726 ;* THIS SUBROUTINE IS USED WHEN A NON-TEST RELATED ERROR HAS BEEN FOUND
3727 ;* DURING THE EXECUTION OF THE CURRENT TEST.
3728 ;* IT IS USED TO INFORM THE OPERATOR THAT THE CURRENT TEST HAS BEEN
3729 ;* ABORTED.
3730 ;*
3731 ;* INPUTS: ERRMSG - CONTAINS THE NAME OF THE CURRENT TEST.
3732 ;*          ERRNR - CONTAINS THE CORRECT ERROR NUMBER.
3733 ;*          THE REMAINDER OF THE ERRBL IS CORRECTLY INITIALISED.
3734 ;*
3735 ;* OUTPUTS: MESSAGES ARE REPORTED TO THE OPERATOR.
3736 ;*
3737 ;* CALLING SEQUENCE: JSR PC,TSABRT
3738 ;*
3739 ;* COMMENTS:
3740 ;*
3741 ;* SUBORDINATE ROUTINES CALLED: ER1603.
3742 ;* *****
3743
3744 022426 TSABRT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3745 022426 004567 161340 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3746 022432 012701 022450 MOV #2,R1 ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
3747 022436 012767 016462 161324 MOV #ER1603,ERRBLK ;SET-UP THE ERROR REPORTING ROUTINE.
3748 022444 104460 ERROR ;
3749 022446 000432 ; >>>> ERROR <<<<. TRAP C#ERROR
3750 022450 040 116 117 2#: BR 60# ;
3751 022453 116 055 122 .ASCIZ / NON-RELATED TEST ERROR FOUND DURING TEST EXECUTION/
3752 022456 105 114 101
3753 022461 124 105 104
3754 022464 040 124 105
3755 022467 123 124 040
3756 022472 105 122 122
3757 022475 117 122 040
3758 022500 106 117 125
3759 022503 116 104 040
3760 022506 104 125 122
3761 022511 111 116 107
3762 022514 040 124 105
3763 022517 123 124 040
3764 022522 105 130 105
3765 022525 103 125 124
3766 022530 111 117 116
3767 022533 000
3768
3769 .EVEN
3770 60#: PASS
3771 022534 004736 JSR ;RESTORE GPRS.
3772 022536 000207 RTS PC ;RETURN TO PREG05 SUBRT.

```

```

3754 .SBTTL GLOBAL SUBROUTINE - TXDSBL -
3755 ;* *****
3756 ;* - TRANSMITTER DISABLE -
3757 ;* THIS SUBROUTINE IS USED TO DISABLE TRANSMISSION ON SELECTED LINES BY,
3758 ;* CLEARING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
3759 ;*
3760 ;* INPUTS: R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO CLEAR TX.ENABLE.
3761 ;* CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
3762 ;* IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
3763 ;* NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
3764 ;* TXAD2A - CONTAINS THE ADDRESS OF THE TBUFAD2 REGISTER.
3765 ;*
3766 ;* OUTPUTS: R5 - BIT'S SET INDICATE THE INITIAL STATES OF ALL TX.ENABLE BITS.
3767 ;* TBUFAD2 - THE STATE OF THE TX.ENABLE BIT MAY BE ALTERED.
3768 ;* THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
3769 ;*
3770 ;* CALLING SEQUENCE: JSR PC,TXDSBL
3771 ;*
3772 ;* COMMENTS:
3773 ;*
3774 ;* SUBORDINATE ROUTINES CALLED: NONE.
3775 ;* -- *****
3776
3777 022540 TXDSBL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3778 022540 004567 161226 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3779 022544 010500 ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
3780 022546 012701 000001 MOV R5,R0 ;INITIALIZE THE SELECTED LINE BIT MASK.
3781 022552 016702 157474 MOV #BIT0,R1 ;GET THE ADDRESS OF THE TBUFAD2 REGISTER.
3782 022556 005202 INC TXAD2A,R2 ;GET THE ADDRESS OF THE MSBYTE OF TBUFAD2 REG.
3783 022560 012703 000020 MOV R2 ;GET MAXIMUM LINE NUMBER PLUS ONE.
3784 022564 016704 157514 MOV #NUMLNS,R3 ;GET THE STATES OF THE INT ENABLE BITS.
3785 022570 005005 MOV IESTAT,R4 ;LOG POSSIBLE TX DISABLED ON ALL LINES.
3786 CLR R5
3787 ;*
3788 ;* SELECT EVERY LINE IN TURN, AND LOG THE STATE OF EACH TX.ENABLE BIT.
3789 ;*
3790 022572 010477 157440 20: MOV R4,BCSRA ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
3791 022576 105712 TSTB (R2) ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
3792 022600 100001 BPL 40 ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
3793 022602 050105 BIS R1,R5 ;LOG TX ENABLE BIT SET FOR SELECTED LINE.
3794 ;*
3795 ;* CLEAR TX.ENABLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX DISABLE
3796 ;* LINE BIT MAP.
3797 ;*
3798 022604 030100 40: BIT R1,R0 ;CHECK STATE OF DISABLE LINE BIT MAP.
3799 022606 001402 BEQ 60 ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
3800 022610 142712 000200 BICB #BIT7,(R2) ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
3801 022614 005204 60: INC R4 ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
3802 022616 006301 ASL R1 ;SHIFT BIT MAP FOR NEXT LINE.
3803 022620 005303 DEC R3 ;DECREMENT LINE NUMBER.
3804 022622 001363 BNE 20 ;LOOP TO CHECK NEXT LINE.
3805 022624 010566 000014 60: PASS R5 ;RESTORE GPRS,EXCEPT
3806 022630 004736 MOV R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
3807 RTS PC JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
3808 ;R5 - PREVIOUS STATES OF ALL TX.ENABLE BITS.

```

```

3808 .SBTTL GLOBAL SUBROUTINE - TXENBL -
3809 ;* *****
3810 ;* - TRANSMITTER ENABLE -
3811 ;* THIS SUBROUTINE IS USED TO ENABLE TRANSMISSION ON SELECTED LINES BY
3812 ;* SETTING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
3813 ;*
3814 ;* INPUTS: R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO SET TX.ENABLE.
3815 ;* CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
3816 ;* IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
3817 ;* NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
3818 ;* TXAD2A - CONTAINS THE ADDRESS OF THE TBUFAD2 REGISTER.
3819 ;*
3820 ;* OUTPUTS: R5 - BIT'S SET INDICATE PREVIOUSLY DISABLED LINES.
3821 ;* TBUFAD2 - THE STATE OF THE TX.ENABLE BIT MAY BE ALTERED.
3822 ;* THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
3823 ;*
3824 ;* CALLING SEQUENCE: JSR PC,TXENBL
3825 ;*
3826 ;* COMMENTS:
3827 ;*
3828 ;* SUBORDINATE ROUTINES CALLED: NONE.
3829 ;* - *****
3830
3831 022634 TXENBL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3832 022634 004567 161132 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3833 022640 010500 MOV R5,R0 ;COPY BIT MAP OF LINES TO ENABLE.
3834 022642 01271 000001 MOV #BIT0,R1 ;INITIALIZE THE SELECTED LINE BIT MASK.
3835 022646 016702 157400 MOV TXAD2A,R2 ;GET THE ADDRESS OF THE TBUFAD2 REGISTER.
3836 022652 005202 INC R2 ;GET THE ADDRESS OF THE MSBYTE OF TBUFAD2 REG.
3837 022654 012703 000020 MOV #NUMLNS,R3 ;GET MAXIMUM LINE NUMBER.
3838 022660 016704 157420 MOV IESTAT,R4 ;GET THE STATES OF THE INT ENABLE BITS.
3839 022664 005005 CLR R5 ;CLEAR TX.ENABLE BIT LOG OF DISABLED LINES.
3840 ;*
3841 ;* SELECT EVERY LINE IN TURN,AND LOG ANY TX.ENABLE BIT THAT IS CLEAR.
3842 022666 010477 157344 20: MOV R4,BCSRA ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
3843 022672 105712 TSTB (R2) ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
3844 022674 100401 BMI 40 ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
3845 022676 050105 BIS R1,R5 ;LOG TX ENABLE BIT CLEAR FOR SELECTED LINE.
3846 ;*
3847 ;* SET TX.ENABLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX ENABLE
3848 ;* LINE BIT MAP.
3849 ;*
3850 022700 030100 40: BIT R1,R0 ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
3851 022702 001402 BEQ 60 ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
3852 022704 152712 000200 BISB #BIT7,(R2) ;ENABLE TRANSMISSION ON SELECTED LINE.
3853 022710 005204 60: INC R4 ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
3854 022712 006301 ASL R1 ;SHIFT BIT MAP FOR NEXT LINE.
3855 022714 005303 DEC R3 ;DECREMENT LINE NUMBER.
3856 022716 001363 BNE 20 ;LOOP TO CHECK NEXT LINE.
3857 ;*
3858 022720 010566 000014 60: PASS R5 ;RESTORE GPRS,EXCEPT
3859 022720 004736 MOV R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
3860 JSR PC,0(SP) ;RETURN TO PREG05 SUBRT.
3861 022726 000207 RTS PC ;R5 - LINE BIT MAP CORRESPONDING TO THE
; PREVIOUS LINES THAT WERE DISABLED.

```

```

3863 .SBTTL GLOBAL SUBROUTINE - TXIE0 -
3864 ; * *****
3865 ; * - TRANSMITTER INTERRUPT DISABLE -
3866 ; * THIS ROUTINE IS USED TO DISABLE TRANSMITTER INTERRUPTS IN THE DHU11.
3867 ; *
3868 ; * INPUTS: NONE.
3869 ; *
3870 ; * OUTPUTS: THE TX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
3871 ; * IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3872 ; * ENABLE BITS.
3873 ; *
3874 ; * CALLING SEQUENCE: JSR PC,TXIE0
3875 ; *
3876 ; * COMMENTS: THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3877 ; * THE DUT CSR ARE DESTROYED.
3878 ; *
3879 ; * SUBORDINATE ROUTINES CALLED: NONE.
3880 ; - *****
3881 022730 010046 TXIE0: MOV RO,-(SP) ;SAVE CONTENTS OF RO ON THE STACK.
3882 022732 GETPRI -(SP) ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
3883 022734 010046 TRAP C#GPRI
3883 022736 SETPRI #PRI07 ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
3884 022742 012700 000340 MOV RO,-(SP)
3884 022744 042767 177677 157332 TRAP C#SPRI
3885 022752 016777 157326 BIC #177677,IESTAT ;CLEAR TX.INT.ENBL BIT IN IESTAT.
3886 022760 MOV IESTAT,0CSRA ;DISABLE TX INTERRUPTS.
3887 022762 012600 SETPRI (SP)+ ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
3888 022764 000207 MOV (SP)+,RO
3888 022766 RTS PC ;RESTORE RO.

```

```

3890      .SBTTL GLOBAL SUBROUTINE                      - TXIE1 -
3891      ;* *****
3892      ;* - TRANSMITTER INTERRUPT ENABLE -
3893      ;* THIS ROUTINE IS USED TO ENABLE TRANSMITTER INTERRUPTS IN THE DHU11.
3894      ;*
3895      ;* INPUTS:      NONE.
3896      ;*
3897      ;* OUTPUTS:     THE TX.INT.ENBL BIT IS SET IN THE DUT CSR.
3898      ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3899      ;*               ENABLE BITS.
3900      ;*
3901      ;* CALLING SEQUENCE:  JSR      PC,TXIE1
3902      ;*
3903      ;* COMMENTS:      THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3904      ;*                  THE DUT CSR ARE DESTROYED.
3905      ;*
3906      ;* SUBORDINATE ROUTINES CALLED: NONE.
3907      ;* - - *****
3908
3909 022770 052767 040000 157306 TXIE1:: BIS    #BIT14,IESTAT ;SET TX.INT.ENBL BIT IN IESTAT.
3910 022776 042767 137677 157300      BIC    #137677,IESTAT ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
3911 023004 016777 157274 157224      MOV    IESTAT,#CSRA  ;ENABLE TX INTERRUPTS.
3912 023012 000207          RTS      PC

```

```
3914 .SBTTL GLOBAL SUBROUTINE - UNSDIV -
3915 ;* *****
3916 ;* - UNSIGNED DIVIDE ROUTINE -
3917 ;* THIS SUBROUTINE IS USED TO DIVIDE A 32 BIT UNSIGNED DIVIDEND BY A
3918 ;* 16 BIT UNSIGNED DIVISOR GIVING A 16 BIT QUOTIENT. ALL NUMBERS ARE
3919 ;* CONSIDERED TO BE UNSIGNED. A SUCCESS FLAG IS NOT SET ON RETURN IF
3920 ;* THE QUOTIENT WAS TOO BIG TO BE CONTAINED IN 16 BITS.
3921 ;*
3922 ;* INPUTS: R1 - THE DIVISOR, UNSIGNED, 16 BITS.
3923 ;* R2 - MOST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
3924 ;* R3 - LEAST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
3925 ;*
3926 ;* OUTPUTS: R1 - QUOTIENT, UNSIGNED, 16 BITS (177777 IF OVERFLOW).
3927 ;* CARRY - SUCCESS FLAG, SET IF COMPLETE QUOTIENT FITS IN 16 BITS.
3928 ;*
3929 ;* CALLING SEQUENCE: JSR PC,UNSDIV
3930 ;*
3931 ;* COMMENTS: IF THE DIVISOR IS 0 THE QUOTIENT IS RETURNED AS ALL ONES
3932 ;* (177777) AND THE CARRY IS CLEAR REGARDLESS OF THE DIVIDEND.
3933 ;*
3934 ;* SUBORDINATE ROUTINES CALLED: NONE.
3935 ;* - *****
3936
3937 023014 UNSDIV:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
023014 004567 160752 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3938
3939 ;*
3940 ;* CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
3941 ;*
3941 023020 010204 MOV R2,R4 ;GET MSW OF DIVIDEND FOR SUBTRACT.
3942 023022 160104 SUB R1,R4 ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
3943 023024 103403 BCS 24 ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
3944 023026 012701 177777 MOV #1,R1 ;SET QUOTIENT TO ALL ONES (177777).
3945 023032 000442 BR 60 ;EXIT WITH CARRY CLEAR.
3946
3947 ;*
3948 ;* SET UP COUNTERS AND VARIOUS WORKING GPRS.
3949 023034 005004 24: CLR R4 ;CLEAR THE LSW OF THE DIVISOR.
3950 023036 000241 CLC ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
3951 023040 006001 ROR R1 ;DIVISOR BY
3952 023042 006004 ROR R4 ;2(UNSIGNED)
3953 023044 012700 000020 MOV #16,,R0 ;SET UP INITIAL SHIFT COUNT TO 16.
3954
3955 ;*
3956 ;* THE SUBTRACT AND SHIFT LOOP.
3957 023050 010246 44: MOV R2,-(SP) ;SAVE MSWORD OF DIVIDEND.
3958 023052 010346 MOV R3,-(SP) ;SAVE LSWORD OF DIVIDEND.
3959 023054 160403 SUB R4,R3 ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
3960 023056 005602 SBC R2 ;MSWORD DIVIDEND - BORROW.
3961 023060 103402 BCS 64 ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
3962 023062 160102 SUB R1,R2 ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
3963 023064 103003 BCC 84 ;IF NO BORROW, IT WENT, CARRY IS CLEAR.
3964
3965 ;*
3966 ;* IT DIDN'T GO, SO WE SHIFT A 1 INTO THE QUOTIENT (COMPLEMENTED LATER).
3967 ;* CARRY IS SET.
3968 023066 012603 64: MOV (SP)+,R3 ;RESTORE LSWORD OF DIVIDEND.
3969 023070 012602 MOV (SP)+,R2 ;RESTORE MSWORD OF DIVIDEND.
```

```
3970 023072 000401      BR      10$      ;GOTO SHIFT 1 INTO THE QUOTIENT.
3971                    ;*
3972                    ; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
3973                    ; COMPLEMENTED LATER).  CARRY IS CLEAR.
3974                    ;
3975 023074 012626      8$:  MOV      (SP)+,(SP)+      ;POP THE SAVED DIVIDEND OFF OF THE STACK.
3976                    ;*
3977                    ; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.
3978                    ;
3979 023076 006105      10$:  ROL      R5              ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
3980 023100 000241      CLC                      ;DIVIDE THE
3981 023102 006001      ROR      R1              ; DIVISOR BY
3982 023104 006004      ROR      R4              ; 2 (UNSIGNED).
3983 023106 005300      DEC      R0              ;COUNT THIS SHIFT AND SUBTRACT.
3984 023110 001357      BNE      4$              ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
3985 023112 005105      COM      R5              ;GET QUOTIENT FROM INVERTED QUOTIENT.
3986                    ;*
3987                    ; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.
3988                    ;
3989 023114 000241      CLC                      ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
3990 023116 006103      ROL      R3              ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
3991 023120 103402      BCS      12$              ;IF CARRY FROM SHIFT, ROUND UP.
3992 023122 160403      SUB      R4,R3              ;SUBTRACT DIVISOR FROM DIVIDEND.
3993 023124 103403      BCS      14$              ;IF BORROW, DON'T ROUND UP.
3994                    ;*
3995                    ; ROUND UP,,EXTRA SUBTRACT WENT.
3996                    ;
3997 023126 005205      12$:  INC      R5              ;INCREMENT THE QUOTIENT BY ONE.
3998 023130 001001      BNE      14$              ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
3999 023132 005305      DEC      R5              ;DON'T LET ROUNDING CAUSE OVERFLOW.
4000                    ;*
4001                    ; ALL DONE, PASS QUOTIENT AND EXIT.
4002                    ;
4003 023134 010501      14$:  MOV      R5,R1              ;PASS QUOTIENT BACK IN R1.
4004 023136 000261      SEC                      ;INDICATE NO OVERFLOW.
4005                    ;
4006 023140                    60$:  PASS      R1              ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
4007 023140 010166 000004      MOV      R1,R1SLOT(SP)      ;PUT R1 IN STACK SLOT.
4008 023144 004736      JSR                      ;RETURN TO PREGOS SUBRT.
4009                    ;R1 - 16 BIT, UNSIGNED QUOTIENT.
4010                    ;CARRY - SET INDICATES NO OVERFLOW (SUCCESS).
4011                    ;
4012                    ;
4013                    ;
4014                    ;
4015                    ;
4016                    ;
4017                    ;
4018                    ;
4019                    ;
4020                    ;
4021                    ;
4022                    ;
4023                    ;
4024                    ;
4025                    ;
4026                    ;
4027                    ;
4028                    ;
4029                    ;
4030                    ;
4031                    ;
4032                    ;
4033                    ;
4034                    ;
4035                    ;
4036                    ;
4037                    ;
4038                    ;
4039                    ;
4040                    ;
4041                    ;
4042                    ;
4043                    ;
4044                    ;
4045                    ;
4046                    ;
4047                    ;
4048                    ;
4049                    ;
4050                    ;
4051                    ;
4052                    ;
4053                    ;
4054                    ;
4055                    ;
4056                    ;
4057                    ;
4058                    ;
4059                    ;
4060                    ;
4061                    ;
4062                    ;
4063                    ;
4064                    ;
4065                    ;
4066                    ;
4067                    ;
4068                    ;
4069                    ;
4070                    ;
4071                    ;
4072                    ;
4073                    ;
4074                    ;
4075                    ;
4076                    ;
4077                    ;
4078                    ;
4079                    ;
4080                    ;
4081                    ;
4082                    ;
4083                    ;
4084                    ;
4085                    ;
4086                    ;
4087                    ;
4088                    ;
4089                    ;
4090                    ;
4091                    ;
4092                    ;
4093                    ;
4094                    ;
4095                    ;
4096                    ;
4097                    ;
4098                    ;
4099                    ;
4100                    ;
4101                    ;
4102                    ;
4103                    ;
4104                    ;
4105                    ;
4106                    ;
4107                    ;
4108                    ;
4109                    ;
4110                    ;
4111                    ;
4112                    ;
4113                    ;
4114                    ;
4115                    ;
4116                    ;
4117                    ;
4118                    ;
4119                    ;
4120                    ;
4121                    ;
4122                    ;
4123                    ;
4124                    ;
4125                    ;
4126                    ;
4127                    ;
4128                    ;
4129                    ;
4130                    ;
4131                    ;
4132                    ;
4133                    ;
4134                    ;
4135                    ;
4136                    ;
4137                    ;
4138                    ;
4139                    ;
4140                    ;
4141                    ;
4142                    ;
4143                    ;
4144                    ;
4145                    ;
4146                    ;
4147                    ;
4148                    ;
4149                    ;
4150                    ;
4151                    ;
4152                    ;
4153                    ;
4154                    ;
4155                    ;
4156                    ;
4157                    ;
4158                    ;
4159                    ;
4160                    ;
4161                    ;
4162                    ;
4163                    ;
4164                    ;
4165                    ;
4166                    ;
4167                    ;
4168                    ;
4169                    ;
4170                    ;
4171                    ;
4172                    ;
4173                    ;
4174                    ;
4175                    ;
4176                    ;
4177                    ;
4178                    ;
4179                    ;
4180                    ;
4181                    ;
4182                    ;
4183                    ;
4184                    ;
4185                    ;
4186                    ;
4187                    ;
4188                    ;
4189                    ;
4190                    ;
4191                    ;
4192                    ;
4193                    ;
4194                    ;
4195                    ;
4196                    ;
4197                    ;
4198                    ;
4199                    ;
4200                    ;
4201                    ;
4202                    ;
4203                    ;
4204                    ;
4205                    ;
4206                    ;
4207                    ;
4208                    ;
4209                    ;
4210                    ;
4211                    ;
4212                    ;
4213                    ;
4214                    ;
4215                    ;
4216                    ;
4217                    ;
4218                    ;
4219                    ;
4220                    ;
4221                    ;
4222                    ;
4223                    ;
4224                    ;
4225                    ;
4226                    ;
4227                    ;
4228                    ;
4229                    ;
4230                    ;
4231                    ;
4232                    ;
4233                    ;
4234                    ;
4235                    ;
4236                    ;
4237                    ;
4238                    ;
4239                    ;
4240                    ;
4241                    ;
4242                    ;
4243                    ;
4244                    ;
4245                    ;
4246                    ;
4247                    ;
4248                    ;
4249                    ;
4250                    ;
4251                    ;
4252                    ;
4253                    ;
4254                    ;
4255                    ;
4256                    ;
4257                    ;
4258                    ;
4259                    ;
4260                    ;
4261                    ;
4262                    ;
4263                    ;
4264                    ;
4265                    ;
4266                    ;
4267                    ;
4268                    ;
4269                    ;
4270                    ;
4271                    ;
4272                    ;
4273                    ;
4274                    ;
4275                    ;
4276                    ;
4277                    ;
4278                    ;
4279                    ;
4280                    ;
4281                    ;
4282                    ;
4283                    ;
4284                    ;
4285                    ;
4286                    ;
4287                    ;
4288                    ;
4289                    ;
4290                    ;
4291                    ;
4292                    ;
4293                    ;
4294                    ;
4295                    ;
4296                    ;
4297                    ;
4298                    ;
4299                    ;
4300                    ;
4301                    ;
4302                    ;
4303                    ;
4304                    ;
4305                    ;
4306                    ;
4307                    ;
4308                    ;
4309                    ;
4310                    ;
4311                    ;
4312                    ;
4313                    ;
4314                    ;
4315                    ;
4316                    ;
4317                    ;
4318                    ;
4319                    ;
4320                    ;
4321                    ;
4322                    ;
4323                    ;
4324                    ;
4325                    ;
4326                    ;
4327                    ;
4328                    ;
4329                    ;
4330                    ;
4331                    ;
4332                    ;
4333                    ;
4334                    ;
4335                    ;
4336                    ;
4337                    ;
4338                    ;
4339                    ;
4340                    ;
4341                    ;
4342                    ;
4343                    ;
4344                    ;
4345                    ;
4346                    ;
4347                    ;
4348                    ;
4349                    ;
4350                    ;
4351                    ;
4352                    ;
4353                    ;
4354                    ;
4355                    ;
4356                    ;
4357                    ;
4358                    ;
4359                    ;
4360                    ;
4361                    ;
4362                    ;
4363                    ;
4364                    ;
4365                    ;
4366                    ;
4367                    ;
4368                    ;
4369                    ;
4370                    ;
4371                    ;
4372                    ;
4373                    ;
4374                    ;
4375                    ;
4376                    ;
4377                    ;
4378                    ;
4379                    ;
4380                    ;
4381                    ;
4382                    ;
4383                    ;
4384                    ;
4385                    ;
4386                    ;
4387                    ;
4388                    ;
4389                    ;
4390                    ;
4391                    ;
4392                    ;
4393                    ;
4394                    ;
4395                    ;
4396                    ;
4397                    ;
4398                    ;
4399                    ;
4400                    ;
4401                    ;
4402                    ;
4403                    ;
4404                    ;
4405                    ;
4406                    ;
4407                    ;
4408                    ;
4409                    ;
4410                    ;
4411                    ;
4412                    ;
4413                    ;
4414                    ;
4415                    ;
4416                    ;
4417                    ;
4418                    ;
4419                    ;
4420                    ;
4421                    ;
4422                    ;
4423                    ;
4424                    ;
4425                    ;
4426                    ;
4427                    ;
4428                    ;
4429                    ;
4430                    ;
4431                    ;
4432                    ;
4433                    ;
4434                    ;
4435                    ;
4436                    ;
4437                    ;
4438                    ;
4439                    ;
4440                    ;
4441                    ;
4442                    ;
4443                    ;
4444                    ;
4445                    ;
4446                    ;
4447                    ;
4448                    ;
4449                    ;
4450                    ;
4451                    ;
4452                    ;
4453                    ;
4454                    ;
4455                    ;
4456                    ;
4457                    ;
4458                    ;
4459                    ;
4460                    ;
4461                    ;
4462                    ;
4463                    ;
4464                    ;
4465                    ;
4466                    ;
4467                    ;
4468                    ;
4469                    ;
4470                    ;
4471                    ;
4472                    ;
4473                    ;
4474                    ;
4475                    ;
4476                    ;
4477                    ;
4478                    ;
4479                    ;
4480                    ;
4481                    ;
4482                    ;
4483                    ;
4484                    ;
4485                    ;
4486                    ;
4487                    ;
4488                    ;
4489                    ;
4490                    ;
4491                    ;
4492                    ;
4493                    ;
4494                    ;
4495                    ;
4496                    ;
4497                    ;
4498                    ;
4499                    ;
4500                    ;
4501                    ;
4502                    ;
4503                    ;
4504                    ;
4505                    ;
4506                    ;
4507                    ;
4508                    ;
4509                    ;
4510                    ;
4511                    ;
4512                    ;
4513                    ;
4514                    ;
4515                    ;
4516                    ;
4517                    ;
4518                    ;
4519                    ;
4520                    ;
4521                    ;
4522                    ;
4523                    ;
4524                    ;
4525                    ;
4526                    ;
4527                    ;
4528                    ;
4529                    ;
4530                    ;
4531                    ;
4532                    ;
4533                    ;
4534                    ;
4535                    ;
4536                    ;
4537                    ;
4538                    ;
4539                    ;
4540                    ;
4541                    ;
4542                    ;
4543                    ;
4544                    ;
4545                    ;
4546                    ;
4547                    ;
4548                    ;
4549                    ;
4550                    ;
4551                    ;
4552                    ;
4553                    ;
4554                    ;
4555                    ;
4556                    ;
4557                    ;
4558                    ;
4559                    ;
4560                    ;
4561                    ;
4562                    ;
4563                    ;
4564                    ;
4565                    ;
4566                    ;
4567                    ;
4568                    ;
4569                    ;
4570                    ;
4571                    ;
4572                    ;
4573                    ;
4574                    ;
4575                    ;
4576                    ;
4577                    ;
4578                    ;
4579                    ;
4580                    ;
4581                    ;
4582                    ;
4583                    ;
4584                    ;
4585                    ;
4586                    ;
4587                    ;
4588                    ;
4589                    ;
4590                    ;
4591                    ;
4592                    ;
4593                    ;
4594                    ;
4595                    ;
4596                    ;
4597                    ;
4598                    ;
4599                    ;
4600                    ;
4601                    ;
4602                    ;
4603                    ;
4604                    ;
4605                    ;
4606                    ;
4607                    ;
4608                    ;
4609                    ;
4610                    ;
4611                    ;
4612                    ;
4613                    ;
4614                    ;
4615                    ;
4616                    ;
4617                    ;
4618                    ;
4619                    ;
4620                    ;
4621                    ;
4622                    ;
4623                    ;
4624                    ;
4625                    ;
4626                    ;
4627                    ;
4628                    ;
4629                    ;
4630                    ;
4631                    ;
4632                    ;
4633                    ;
4634                    ;
4635                    ;
4636                    ;
4637                    ;
4638                    ;
4639                    ;
4640                    ;
4641                    ;
4642                    ;
4643                    ;
4644                    ;
4645                    ;
4646                    ;
4647                    ;
4648                    ;
4649                    ;
4650                    ;
4651                    ;
4652                    ;
4653                    ;
4654                    ;
4655                    ;
4656                    ;
4657                    ;
4658                    ;
4659                    ;
4660                    ;
4661                    ;
4662                    ;
4663                    ;
4664                    ;
4665                    ;
4666                    ;
4667                    ;
4668                    ;
4669                    ;
4670                    ;
4671                    ;
4672                    ;
4673                    ;
4674                    ;
4675                    ;
4676                    ;
4677                    ;
4678                    ;
4679                    ;
4680                    ;
4681                    ;
4682                    ;
4683                    ;
4684                    ;
4685                    ;
4686                    ;
4687                    ;
4688                    ;
4689                    ;
4690                    ;
4691                    ;
4692                    ;
4693                    ;
4694                    ;
4695                    ;
4696                    ;
4697                    ;
4698                    ;
4699                    ;
4700                    ;
4701                    ;
4702                    ;
4703                    ;
4704                    ;
4705                    ;
4706                    ;
4707                    ;
4708                    ;
4709                    ;
4710                    ;
4711                    ;
4712                    ;
4713                    ;
4714                    ;
4715                    ;
4716                    ;
4717                    ;
4718                    ;
4719                    ;
4720                    ;
4721                    ;
4722                    ;
4723                    ;
4724                    ;
4725                    ;
4726                    ;
4727                    ;
4728                    ;
4729                    ;
4730                    ;
4731                    ;
4732                    ;
4733                    ;
4734                    ;
4735                    ;
4736                    ;
4737                    ;
4738                    ;
4739                    ;
4740                    ;
4741                    ;
4742                    ;
4743                    ;
4744                    ;
4745                    ;
4746                    ;
4747                    ;
4748                    ;
4749                    ;
4750                    ;
4751                    ;
4752                    ;
4753                    ;
4754                    ;
4755                    ;
4756                    ;
4757                    ;
4758                    ;
4759                    ;
4760                    ;
4761                    ;
4762                    ;
4763                    ;
4764                    ;
4765                    ;
4766                    ;
4767                    ;
4768                    ;
4769                    ;
4770                    ;
4771                    ;
4772                    ;
4773                    ;
4774                    ;
4775                    ;
4776                    ;
4777                    ;
4778                    ;
4779                    ;
4780                    ;
4781                    ;
4782                    ;
4783                    ;
4784                    ;
4785                    ;
4786                    ;
4787                    ;
4788                    ;
4789                    ;
4790                    ;
4791                    ;
4792                    ;
4793                    ;
4794                    ;
4795                    ;
4796                    ;
4797                    ;
4798                    ;
4799                    ;
4800                    ;
4801                    ;
4802                    ;
4803                    ;
4804                    ;
4805                    ;
4806                    ;
4807                    ;
4808                    ;
4809                    ;
4810                    ;
4811                    ;
4812                    ;
4813                    ;
4814                    ;
4815                    ;
4816                    ;
4817                    ;
4818                    ;
4819                    ;
4820                    ;
4821                    ;
4822                    ;
4823                    ;
4824                    ;
4825                    ;
4826                    ;
4827                    ;
4828                    ;
4829                    ;
4830                    ;
4831                    ;
4832                    ;
4833                    ;
4834                    ;
4835                    ;
4836                    ;
4837                    ;
4838                    ;
4839                    ;
4840                    ;
4841                    ;
4842                    ;
4843                    ;
4844                    ;
4845                    ;
4846                    ;
4847                    ;
4848                    ;
4849                    ;
4850                    ;
4851                    ;
4852                    ;
4853                    ;
4854                    ;
4855                    ;
4856                    ;
4857                    ;
4858                    ;
4859                    ;
4860                    ;
4861                    ;
4862                    ;
4863                    ;
4864                    ;
4865                    ;
4866                    ;
4867                    ;
4868                    ;
4869                    ;
4870                    ;
4871                    ;
4872                    ;
4873                    ;
4874                    ;
4875                    ;
4876                    ;
4877                    ;
4878                    ;
4879                    ;
4880                    ;
4881                    ;
4882                    ;
4883                    ;
4884                    ;
4885                    ;
4886                    ;
4887                    ;
4888                    ;
4889                    ;
4890                    ;
4891                    ;
4892                    ;
4893                    ;
4894                    ;
4895                    ;
4896                    ;
4897                    ;
4898                    ;
4899                    ;
4900                    ;
4901                    ;
4902                    ;
4903                    ;
4904                    ;
4905                    ;
4906                    ;
4907                    ;
4908                    ;
4909                    ;
4910                    ;
4911                    ;
4912                    ;
4913                    ;
4914                    ;
4915                    ;
4916                    ;
4917                    ;
4918                    ;
4919                    ;
4920                    ;
4921                    ;
4922                    ;
4923                    ;
4924                    ;
4925                    ;
4926                    ;
4927                    ;
4928                    ;
4929                    ;
4930                    ;
4931                    ;
4932                    ;
4933                    ;
4934                    ;
4935                    ;
4936                    ;
4937                    ;
4938                    ;
4939                    ;
4940                    ;
4941                    ;
4942                    ;
4943                    ;
4944                    ;
4945                    ;
4946                    ;
4947                    ;
4948                    ;
4949                    ;
4950                    ;
4951                    ;
4952                    ;
4953                    ;
4954                    ;
4955                    ;
4956                    ;
4957                    ;
4958                    ;
4959                    ;
4960                    ;
4961                    ;
4962                    ;
4963                    ;
4964                    ;
4965                    ;
4966                    ;
4967                    ;
4968                    ;
4969                    ;
4970                    ;
4971                    ;
4972                    ;
4973                    ;
4974                    ;
4975                    ;
4976                    ;
4977                    ;
4978                    ;
4979                    ;
4980                    ;
4981                    ;
4982                    ;
4983                    ;
4984                    ;
4985                    ;
4986                    ;
4987                    ;
4988                    ;
4989                    ;
4990                    ;
4991                    ;
4992                    ;
4993                    ;
4994                    ;
4995                    ;
4996                    ;
4997                    ;
4998                    ;
4999                    ;
5000                    ;
5001                    ;
5002                    ;
5003                    ;
5004                    ;
5005                    ;
5006                    ;
5007                    ;
5008                    ;
5009                    ;
5010                    ;
5011                    ;
5012                    ;
5013                    ;
5014                    ;
5015                    ;
5016                    ;
5017                    ;
5018                    ;
5019                    ;
5020                    ;
5021                    ;
5022                    ;
5023                    ;
5024                    ;
5025                    ;
5026                    ;
5027                    ;
5028                    ;
5029                    ;
5030                    ;
5031                    ;
5032                    ;
5033                    ;
5034                    ;
5035                    ;
5036                    ;
5037                    ;
5038                    ;
5039                    ;
5040                    ;
5041                    ;
5042                    ;
5043                    ;
5044                    ;
5045                    ;
5046                    ;
5047                    ;
5048                    ;
5049                    ;
5050                    ;
5051                    ;
5052                    ;
5053                    ;
5054                    ;
5055                    ;
5056                    ;
5057                    ;
5058                    ;
5059                    ;
5060                    ;
5061                    ;
5062                    ;
5063                    ;
5064                    ;
5065                    ;
5066                    ;
5067                    ;
5068                    ;
5069                    ;
5070                    ;
5071                    ;
5072                    ;
5073                    ;
5074                    ;
5075                    ;
5076                    ;
5077                    ;
5078                    ;
5079                    ;
5080                    ;
5081                    ;
5082                    ;
5083                    ;
5084                    ;
5085                    ;
5086                    ;
5087                    ;
5088                    ;
5089                    ;
5090                    ;
5091                    ;
5092                    ;
5093                    ;
5094                    ;
5095                    ;
5096                    ;
5097                    ;
5098                    ;
5099                    ;
5100                    ;
5101                    ;
5102                    ;
5103                    ;
5104                    ;
5105                    ;
5106                    ;
5107                    ;
5108                    ;
5109                    ;
5110                    ;
5111                    ;
5112                    ;
5113                    ;
5114                    ;
5115                    ;
5116                    ;
5117                    ;
5118                    ;
5119                    ;
5120                    ;
5121                    ;
5122                    ;
5123                    ;
5124                    ;
5125                    ;
5126                    ;
5127                    ;
5128                    ;
5129                    ;
5130                    ;
5131                    ;
5132                    ;
5133                    ;
5134                    ;
5135                    ;
5136                    ;
5137                    ;
5138                    ;
5139                    ;
5140                    ;
5141                    ;
5142                    ;
5143                    ;
5144                    ;
5145                    ;
5146                    ;
5147                    ;
5148                    ;
5149                    ;
5150                    ;
5151                    ;
5152                    ;
5153                    ;
5154                    ;
5155                    ;
5156                    ;
5157                    ;
5158                    ;
5159                    ;
5160                    ;
5161                    ;
5162                    ;
5163                    ;
5164                    ;
5165                    ;
5166                    ;
5167                    ;
5168                    ;
5169                    ;
5170                    ;
5171                    ;
5172                    ;
5173                    ;
5174                    ;
5175                    ;
5176                    ;
5177                    ;
5178                    ;
5179                    ;
5180                    ;
5181                    ;
5182                    ;
5183                    ;
5184                    ;
5185                    ;
5186                    ;
5187                    ;
5188                    ;
5189                    ;
5190                    ;
5191                    ;
5192                    ;
5193                    ;
5194                    ;
5195                    ;
5196                    ;
5197                    ;
5198                    ;
5199                    ;
5200                    ;
5201                    ;
5202                    ;
5203                    ;
5204                    ;
5205                    ;
5206                    ;
5207                    ;
5208                    ;
5209                    ;
5210                    ;
5211                    ;
5212                    ;
5213                    ;
5214                    ;
5215                    ;
5216                    ;
5217                    ;
5218                    ;
5219                    ;
5220                    ;
5221                    ;
5222                    ;
5223                    ;
5224                    ;
5225                    ;
5226                    ;
5227                    ;
5228                    ;
5229                    ;
5230                    ;
5231                    ;
5232                    ;
5233                    ;
5234                    ;
5235                    ;
5236                    ;
5237                    ;
5238                    ;
5239                    ;
5240                    ;
5241                    ;
5242                    ;
5243                    ;
5244                    ;
5245                    ;
5246                    ;
5247                    ;
5248                    ;
5249                    ;
5250                    ;
5251                    ;
5252                    ;
5253                    ;
5254                    ;
5255                    ;
5256                    ;
5257                    ;
5258                    ;
5259                    ;
5260                    ;
5261                    ;
5262                    ;
5263                    ;
5264                    ;
5265                    ;
5266                    ;
5267                    ;
5268                    ;
5269                    ;
5270                    ;
5271                    ;
5272                    ;
5273                    ;
5274                    ;
5275                    ;
5276                    ;
5277                    ;
5278                    ;
5279                    ;
5280                    ;
5281                    ;
5282                    ;
5283                    ;
5284                    ;
5285                    ;
5286                    ;
5287                    ;
5288                    ;
5289                    ;
5290                    ;
5291                    ;
5292                    ;
5293                    ;
5294                    ;
5295                    ;
5296                    ;
5297                    ;
5298                    ;
5299                    ;
5300                    ;
5301                    ;
5302                    ;
5303                    ;
5304                    ;
5305                    ;
5306                    ;
5307                    ;
5308                    ;
5309                    ;
5310                    ;
5311                    ;
5312                    ;
5313                    ;
5314                    ;
5315                    ;
5316                    ;
5317                    ;
5318                    ;
5319                    ;
5320                    ;
5321                    ;
5322                    ;
5323                    ;
5324                    ;
5325                    ;
5326                    ;
5327                    ;
5328                    ;
5329                    ;
5330                    ;
5331                    ;
5332                    ;
5333                    ;
5334                    ;
5335                    ;
5336                    ;
5337                    ;
5338                    ;
5339                    ;
5340                    ;
5341                    ;
5342                    ;
5343                    ;
5344                    ;
5345                    ;
5346                    ;
5347                    ;
5348                    ;
5349                    ;
5350                    ;
5351                    ;
5352                    ;
5353                    ;
5354                    ;
5355                    ;
5356                    ;
5357                    ;
5358                    ;
5359                    ;
5360                    ;
5361                    ;
5362                    ;
5363                    ;
5364                    ;
5365                    ;
5366                    ;
5367                    ;
5368                    ;
5369                    ;
5370                    ;
5371                    ;
5372                    ;
5373                    ;
5374                    ;
5375                    ;
5376                    ;
5377                    ;
5378                    ;
5379                    ;
5380                    ;
5381                    ;
5382                    ;
5383                    ;
5384                    ;
5385                    ;
5386                    ;
5387                    ;
5388                    ;
5389                    ;
5390                    ;
5391                    ;
5392                    ;
5393                    ;
5394                    ;
5395                    ;
5396                    ;
5397                    ;
5398                    ;
5399                    ;
5400                    ;
5401                    ;
5402                    ;
5403                    ;
5404                    ;
5405                    ;
5406                    ;
5407                    ;
5408                    ;
5409                    ;
5410                    ;
5411                    ;
5412                    ;
5413                    ;
5414                    ;
5415                    ;
5416                    ;
5417                    ;
5418                    ;
5419                    ;
5420                    ;
5421                    ;
5422                    ;
5423                    ;
5424                    ;
5425                    ;
5426                    ;
5427                    ;
5428                    ;
5429                    ;
5430                    ;
5431                    ;
5432                    ;
5433                    ;
5434                    ;
5435                    ;
5436                    ;
5437                    ;
5438                    ;
5439                    ;
5440                    ;
5441                    ;
5442                    ;
5443                    ;
5444                    ;
5445                    ;
5446                    ;
5447                    ;
5448                    ;
5449                    ;
5450                    ;
5451                    ;
5452                    ;
5453                    ;
5454                    ;
5455                    ;
5456                    ;
5457                    ;
5458                    ;
5459                    ;
5460                    ;
5461                    ;
5462                    ;
5463                    ;
5464                    ;
5465                    ;
5466                    ;
5467                    ;
5468                    ;
5469                    ;
5470                    ;
5471                    ;
5472                    ;
5473                    ;
5474                    ;
5475                    ;
5476                    ;
5477                    ;
5478                    ;
5479                    ;
5480                    ;
5481                    ;
5482                    ;
5483                    ;
5484                    ;
5485                    ;
5486                    ;
5487                    ;
5488                    ;
5489                    ;
5490                    ;
5491                    ;
5492                    ;
5493                    ;
5494                    ;
5495                    ;
5496                    ;
5497                    ;
5498                    ;
5499                    ;
5500                    ;
5501                    ;
5502                    ;
5503                    ;
5504                    ;
5505                    ;
5506                    ;
5507                    ;
5508                    ;
5509                    ;
5510                    ;
5511                    ;
5512                    ;
5513                    ;
5514                    ;
5515                    ;
5516                    ;
5517                    ;
5518                    ;
5519                    ;
5520                    ;
5521                    ;
5522                    ;
5523                    ;
5524                    ;
5525                    ;
5526                    ;
5527                    ;
5528                    ;
5529                    ;
5530                    ;
5531                    ;
5532                    ;
5533                    ;
5534                    ;
5535                    ;
5536                    ;
5537                    ;
5538                    ;
5539                    ;
5540                    ;
5541                    ;
5542                    ;
5543                    ;
5544                    ;
5545                    ;
5546                    ;
5547                    ;
5548                    ;
5549                    ;
5550                    ;
5551                    ;
5552                    ;
5553                    ;
5554                    ;
5555                    ;
5556                    ;
5557                    ;
5558                    ;
5559                    ;
5560                    ;
5561                    ;
5562                    ;
5563                    ;
5564                    ;
5565                    ;
5566                    ;
5567                    ;
5568                    ;
5569                    ;
5570                    ;
5571                    ;
5572                    ;
5573                    ;
5574                    ;
5575                    ;
5576                    ;
5577                    ;
5578                    ;
5579                    ;
5580                    ;
5581                    ;
5582                    ;
5583                    ;
5584                    ;
5585                    ;
5586                    ;
5587                    ;
5588                    ;
5589                    ;
5590                    ;
5591                    ;
5592                    ;
5593                    ;
5594                    ;
5595                    ;
5596                    ;
5597                    ;
5598                    ;
5599                    ;
5600                    ;
5
```

4010 .SBTTL GLOBAL SUBROUTINE - WAIBIS -
4011
4012 ;* - WAIT FOR BIT SET ROUTINE -
4013 ;* THIS SUBROUTINE WAITS FOR THE SPECIFIED BIT TO BECOME SET. IF THE
4014 ;* SPECIFIED BIT GOES TO A SET STATE WITHIN THE SPECIFIED TIME-OUT
4015 ;* PERIOD A SUCCESS INDICATION IS RETURNED BY THIS ROUTINE.
4016 ;* THE LAST VALUE WHICH IS READ LOOKING FOR THE CONDITION IS RETURNED TO
4017 ;* ALLOW THE USE OF THIS ROUTINE TO LOOK FOR DESTRUCTIVE READ CONDITIONS.
4018 ;*
4019 ;* INPUTS: R1 - TIME-OUT VALUE AND BIT NUMBER INDICATION:
4020 ;* BITS 15 THRU 12 - NUMBER OF BIT TO TEST (RANGE 0 THRU 15).
4021 ;* BITS 11 THRU 0 - TIME-OUT VALUE IN MILLI-SECONDS (4095 MAX).
4022 ;* R2 - ADDRESS OF WORD CONTAINING THE BIT TO TEST.
4023 ;* MSLCNT.
4024 ;*
4025 ;* OUTPUTS: R2 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
4026 ;* CARRY - SUCCESS FLAG (CARRY SET IF BIT SET BEFORE TIME-OUT).
4027 ;*
4028 ;* CALLING SEQUENCE: MOV #130040,R1 ;PASS BIT 11 (13 OCTAL) AND
4029 ;* ; 32 (40 OCTAL) MS DELAY.
4030 ;* MOV #LABEL,R2 ;TEST BIT IN WORD AT "LABEL".
4031 ;* JSR PC,WAIBIS ;WAIT 32 MS FOR BIT 11 TO SET.
4032 ;*
4033 ;* COMMENTS:
4034 ;*
4035 ;* SUBORDINATE ROUTINES CALLED: MSLGET.
4036 ;*
4037
4038 023150 WAIBIS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
4039 023150 004567 160616 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4040 023154 010204 MOV R2,R4 ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
4041 023156 010102 MOV R1,R2
4042 023160 042701 170000 BIC #170000,R1 ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
4043 023164 042702 007777 BIC #7777,R2 ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
4044 023170 000302 SWAB R2 ;PUT LINE NUMBER FIELD IN LSBYTE.
4045 023172 006202 ASR R2 ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
4046 023174 006202 ASR R2 ; POSITION TO USE IT AS A WORD TABLE OFFSET
4047 023200 016202 002404 MOV BITTBL(R2),R2 ; FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
4048 023204 010203 MOV R2,R3 ;GET BIT MAP OF LINE TO TEST FROM TABLE.
4049 023206 004767 174706 JSR PC,MSLGET ;INDICATE THAT THE BIT SHOULD BE SET.
4050 ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
4051 023212 010002 MOV R0,R2 ; CARRY IS CORRECT UPON MSLGET RETURN.
4052 023214 010266 000006 601: PASS R2 ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
4053 023220 004736 MOV R2,R2SLOT(SP) ;RESTORE GPRS, EXCEPT THE FOLLOWING:
4054 023222 000207 JSR PC,@(SP) ;PUT R2 IN STACK SLOT.
;RETURN TO PREG05 SUBRT.
; R2 - LAST VALUE READ LOOKING FOR CONDITION.
; CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).

```
4056 .SBTTL GLOBAL SUBROUTINE - WDPDR -
4057 ;* *****
4058 ;* - WRITE DATA PATTERN TO DEVICE REGISTERS -
4059 ;* THIS ROUTINE WRITES A ROTATED DATA PATTERN TO EACH OF THE 6 DEVICE
4060 ;* REGISTERS OF EACH ACTIVE LINE OF THE DEVICE UNDER TEST.
4061 ;* THE DATA PATTERN IS ROTATED ONCE AFTER EACH WRITE TO A DEVICE REGISTER
4062 ;* ON A PARTICULAR LINE. THE STARTING DATA PATTERN FOR EACH LINE
4063 ;* IS ROTATED ONCE AFTER WRITING ALL THE REGISTERS ON A PARTICULAR
4064 ;* LINE. THIS LEADS TO THE FOLLOWING DATA PATTERN:
4065 ;* LINE 0, REGISTER 0 - SHIFTED 0 BIT POSITIONS
4066 ;* LINE 0, REGISTER 1 - SHIFTED 1 BIT POSITION
4067 ;*
4068 ;* LINE 1, REGISTER 0 - SHIFTED 1 BIT POSITION
4069 ;* LINE 2, REGISTER 1 - SHIFTED 2 BIT POSITIONS
4070 ;*
4071 ;* ANY BITS FIELDS IN THE DEVICE REGISTERS THAT CANNOT BE ALTERED
4072 ;* ARE MASKED OUT OF THE DATA PATTERN BEFORE IT IS WRITTEN.
4073 ;* THIS ROUTINE WILL USE EITHER MOV, MOVB, BIS, BISB, BIC, OR BICB
4074 ;* INSTRUCTIONS. THE UPPER OR LOWER BYTE CAN BE SPECIFIED FOR WRITING.
4075 ;*
4076 ;* INPUTS: R2 - USED TO PASS IN THE DATA PATTERN TO BE ROTATED & WRITTEN.
4077 ;* R3 - BYTE INDICATOR (- => LO BYTE, + => HI BYTE, 0 => BOTH).
4078 ;* R4 - OPERATION TYPE INDICATOR (- => BIC, + => BIS, 0 => MOV).
4079 ;* ACTLNS - BIT MAP OF THE ACTIVE LINES ON THE DEVICE UNDER TEST.
4080 ;* CSRA - CONTAINS THE CSR ADDRESS OF THE DEVICE UNDER TEST.
4081 ;* DRADRT - BASE ADDRESS OF DEVICE REGISTER ADDRESS TABLE.
4082 ;* LPRO - EQUATED TO LPR REG OFFSET FROM DEVICE CSR ADDRESS.
4083 ;* NUMLNS - NUMBER OF LINES ON THE DEVICE UNDER TEST.
4084 ;* TXBFCO - EQUATED TO TBUFFT REG OFFSET FROM DEVICE CSR ADDRESS.
4085 ;* UNBITB - BASE ADDRESS OF THE UNUSED BIT TABLE.
4086 ;*
4087 ;* OUTPUTS: DEVICE REGISTERS ON ALL ACTIVE DEVICE LINES ARE MODIFIED.
4088 ;*
4089 ;* CALLING SEQUENCE: JSR PC,WDPDR
4090 ;*
4091 ;* COMMENTS: THIS ROUTINE DOES NOT WRITE DATA TO THE FOLLOWING REGISTERS,
4092 ;* RBUF
4093 ;* RXTIMER
4094 ;* STAT
4095 ;* FIFOSIZE
4096 ;* FIFODATA
4097 ;*
4098 ;* THE CSR IS CLEARED EXCEPT FOR THE IND.ADR.REG FIELD.
4099 ;*
4100 ;* SUBORDINATE ROUTINES CALLED: ROLDAP.
4101 ;* *****
4102 023224 WDPDR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
4103 023224 004567 160542 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4104 ;*
4105 ;* SET UP OUTER LOOP WHICH WRITES THE DATA PATTERN TO EACH LINE'S REGISTERS
4106 023230 005005 ;- CLR R5 ;CLEAR LINE COUNTER TO SELECT LINE 0.
4107 ;*
4108 ;* THE OUTER LOOP FOLLOWS. EACH PASS THROUGH THIS LOOP WRITES DATA TO ALL OF
4109 ;* THE DEVICE REGISTERS FOR A PARTICULAR LINE IF IT IS ACTIVE.
4110 ;*
4111 023232 010204 20: MOV R2,R4 ;SAVE THE OUTER LOOP DATA PATTERN.
```

```

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

```

ADDRESS	OPCODE	OPERANDS	COMMENT
4169	023416	005205	INC R5 ;COUNT THIS LINE.
4170	023420	020527 000020	CMP R5,#NUMLNS ;COMPARE LINE COUNT WITH NUMBER OF LINES.
4171	023424	002702	BLT 2\$;LOOP IF SOME LINES NOT DONE.
4172			
4173	023426	60\$:	PASS ;RESTORE GPRS.
	023426	004736	JSR PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
4174	023430	000207	RTS PC

H9

```
4176 .SBTTL GLOBAL SUBROUTINE - WTWLNC -
4177 ;* *****
4178 ;* - LINE CONTROL REGISTER SETUP ROUTINE -
4179 ;* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
4180 ;* CONTROL REGISTERS (LNCTRL) TO THE SPECIFIED STATE. ONLY THE LNCTRLS
4181 ;* FOR THE SPECIFIED LINES ARE ALTERED.
4182 ;*
4183 ;* INPUTS: R0 - NEW LINE PARAMETERS.
4184 ;* R5 - BIT MAP OF LINES TO BE ALTERED.
4185 ;* CSRA - CONTAINS ADDRESS OF THE DUT CSR.
4186 ;* IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
4187 ;* ENABLE BITS IN THE CSR.
4188 ;* LNCTRA - CONTAINS ADDRESS OF THE DUT LNCTRL REGISTERS.
4189 ;*
4190 ;* OUTPUTS: LNCTRL - SPECIFIED DUT LINE CONTROL REGISTERS ARE ALTERED.
4191 ;*
4192 ;* CALLING SEQUENCE: JSR PC,WTWLNC
4193 ;*
4194 ;* COMMENTS:
4195 ;*
4196 ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
4197 ;* - *****
4198
4199 023432 WTWLNC:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
023432 004567 160334 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4200
4201 ;*
4202 ;* SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
4203 023436 016701 156604 ;*
4204 023442 010002 ;*
4205 023444 010503 ;*
4206 023446 012704 177777 ;*
4207 ;*
4208 ;* MOV LNCTRA,R1 ;SET UP THE REGISTER ADDRESS PARAMETER.
4209 ;* MOV R0,R2 ;SET UP THE DESIRED REGISTER CONTENTS.
4210 023452 004767 173732 ;*
4211 ;* MOV R5,R3 ;SET UP THE BIT MAP OF LINES TO ALTER.
4212 023456 004736 ;*
4213 023460 000207 ;*
;* MOV #-1,R4 ;SELECT ALL REGISTER BITS TO BE ALTERED.
;*
;* CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
;*
;* JSR PC,ALTFLD ;ALTER THE REGISTER CONTENTS.
600: PASS ;RESTORE GPRS.
JSR PC,B(SP), ;RETURN TO PREG05 SUBRT.
RTS PC
```

```

4215 .SBTTL GLOBAL SUBROUTINE - WTWLPR -
4216 ;* *****
4217 ;* - LINE PARAMETER REGISTER SETUP ROUTINE -
4218 ;* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
4219 ;* PARAMETER REGISTERS (LPR) TO THE SPECIFIED STATE. ONLY THE LPRS FOR
4220 ;* THE SPECIFIED LINES ARE ALTERED.
4221 ;*
4222 ;* INPUTS: R0 - NEW LINE PARAMETERS.
4223 ;* R5 - BIT MAP OF LINES TO BE ALTERED.
4224 ;* CSRA - CONTAINS ADDRESS OF THE DUT CSR.
4225 ;* IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
4226 ;* ENABLE BITS IN THE CSR.
4227 ;* LPRA - CONTAINS ADDRESS OF THE DUT LPR.
4228 ;*
4229 ;* OUTPUTS: LPR - SPECIFIED DUT LINE PARAMETER REGISTERS ARE ALTERED.
4230 ;*
4231 ;* CALLING SEQUENCE: JSR PC,WTWLPR
4232 ;*
4233 ;* COMMENTS:
4234 ;*
4235 ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
4236 ;* - *****
4237
4238 023462 WTWLPR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
4239 023462 004567 160304 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4240 ;*
4241 ;* SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
4242 ;* -
4242 023466 016701 156550 MOV LPRA,R1 ;SET UP THE REGISTER ADDRESS PARAMETER.
4243 023472 010002 MOV R0,R2 ;SET UP THE DESIRED REGISTER CONTENTS.
4244 023474 010503 MOV R5,R3 ;SET UP THE BIT MAP OF LINES TO ALTER.
4245 023476 012704 177777 MOV #-1,R4 ;SELECT ALL REGISTER BITS TO BE ALTERED.
4246 ;*
4247 ;* CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
4248 ;* -
4249 023502 004767 173702 JSR PC,ALTFLD ;ALTER THE REGISTER CONTENTS.
4250 ;*
4251 023506 004736 604: PASS ;RESTORE GPRS.
4252 023510 000207 RTS PC JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.

```

```

4254 .SBTTL INTERRUPT SERVICE ROUTINE - CACHRX -
4255 ;* *****
4256 ;* - CATCH RECEIVER INTERRUPT.
4257 ;* THIS ROUTINE IS USED IN SEVERAL TESTS, TO LOG A COUNT OF THE
4258 ;* NUMBER OF RECEIVER INTERRUPTS THAT OCCUR.
4259 ;*
4260 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR.
4261 ;* RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERRUPTS
4262 ;* THAT OCCURRED.
4263 ;*
4264 ;* OUTPUTS: RXINTC - CONTAINS THE UPDATED INTERRUPT COUNT.
4265 ;*
4266 ;*
4267 ;* CALLING SEQUENCE: PUT THE ADDRESS OF THE LABEL CACHRX IN THE VECTOR
4268 ;* LOCATION.
4269 ;*
4270 ;* COMMENTS:
4271 ;*
4272 ;* SUBORDINATE ROUTINES CALLED: NONE
4273 ;* -- *****
4274
4275 023512 CACHRX::SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
      023512 004567 160254 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4276 023516 016701 156566 ;GET THE RECEIVER INTERRUPT COUNT
      MOV RXINTC,R1 ;INCREMENT THE COUNT
4277 023522 005201 ;BRANCH IF NO OVERFLOW OCCURRED
      INC R1 ;RESET THE COUNT TO 177777
4278 023524 102001 ;SAVE NEW COUNT VALUE
      BVC 2$ ;RESTORE GPRS.
4279 023526 005301 ;RETURN TO PREG05 SUBRT.
4280 023530 010167 156554 2$: MOV R1,RXINTC
4281 023534 004736 60$: PASS
      JSR PC,B(SP)+
4282 023536 000002 RTI

```

1<9

```
4284 .SBTTL INTERRUPT SERVICE ROUTINE - CACHTX -
4285 ;* *****
4286 ;* - CATCH TRANSMITTER INTERRUPT.
4287 ;* THIS ROUTINE IS USED IN SEVERAL TESTS, TO LOG A COUNT OF THE
4288 ;* NUMBER OF TRANSMISSION INTERRUPTS THAT OCCUR.
4289 ;*
4290 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR.
4291 ;* TXINTC - HOLDS THE COUNT OF THE NUMBER OF TX INTERRUPTS
4292 ;* THAT OCCURRED.
4293 ;*
4294 ;* OUTPUTS: TXINTC - CONTAINS THE UPDATED INTERRUPT COUNT.
4295 ;*
4296 ;*
4297 ;* CALLING SEQUENCE: PUT THE ADDRESS OF THE LABEL CACHTX IN THE VECTOR
4298 ;* LOCATION.
4299 ;*
4300 ;* COMMENTS:
4301 ;*
4302 ;* SUBORDINATE ROUTINES CALLED: NONE
4303 ;* -- *****
4304
4305 023540 CACHTX::SAVE
4306 023540 004567 160226 JSR R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
4307 023550 005201 156552 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4308 023552 102001 ;GET THE TRANSMISSION INTERRUPT COUNT
4309 023554 005301 ;INCREMENT THE COUNT
4310 023556 010167 156540 BVC 2$ ;BRANCH IF NO OVERFLOW OCCURRED
4311 023562 DEC R1 ;RESET THE COUNT TO 177777
4312 023564 004736 000002 2$: MOV R1,TXINTC ;SAVE NEW COUNT VALUE
60$: PASS ;RESTORE GPRS.
RTI JSR PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
```

```

4314 .SBTTL INTERRUPT SERVICE ROUTINE - CLKINT -
4315 ;* *****
4316 ;* THIS ROUTINE IS EXECUTED CLKHRZ TIMES PER SECOND. IT DECREMENTS THE
4317 ;* TWO TIMER COUNTERS DOWN TO ZERO.
4318 ;*
4319 ;* INPUTS: TIMER1 - TIMER COUNTER #1.
4320 ;*          TIMER2 - TIMER COUNTER #2.
4321 ;*          TIMER3 - TIMER COUNTER FOR CALL OF BREAK MACRO.
4322 ;*
4323 ;* OUTPUTS: THE 2 TIMER COUNTERS ARE DECREMENTED IF THEY ARE NOT ZERO.
4324 ;*
4325 ;* CALLING SEQUENCE: PUT #CLKINT IN THE CLOCK INTERRUPT VECTOR SLOT.
4326 ;*                   PUT THE DESIRED TIME PERIOD (SECONDS TIMES CLKHRZ) IN
4327 ;*                   EITHER TIMER1 OR TIMER2 AND POLL THE RESPECTIVE TIMER
4328 ;*                   COUNTER TO DETECT ITS GOING TO 0 ON TIME-OUT.
4329 ;*
4330 ;* COMMENTS: THE 2 COUNTERS WILL NOT WRAPAROUND BUT WILL STOP AT 0. THIS
4331 ;*            ALLOWS THE DETECTION OF A TIME-OUT ANY TIME AFTER THE TIME-OUT
4332 ;*            HAS OCCURRED UNTIL THE TIMER COUNTER IS SET TO ANOTHER VALUE.
4333 ;*
4334 ;* SUBORDINATE ROUTINES CALLED: NONE.
4335 ;* -- *****
4336
4337 023566 005767 156546 CLKINT:: TST    TIMER1    ;CHECK FOR TIMER1 AT ZERO.
4338 023572 001402      BEQ     2$      ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
4339 023574 005367 156540      DEC     TIMER1    ;DECREMENT TIME COUNT.
4340 023600 005767 156536 2$: TST    TIMER2    ;CHECK FOR TIMER2 AT ZERO.
4341 023604 001402      BEQ     4$      ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
4342 023606 005367 156530      DEC     TIMER2    ;DECREMENT TIME COUNT.
4343 023612 005367 156526 4$: DEC     TIMER3    ;DECREMENT THE BREAK COUNT.
4344 023616 001006      BNE     60$      ;EXIT IF NOT TIME TO CALL BREAK.
4345 023620 016767 156522 156516 MOV    BCOUNT,TIMER3 ;SET UP TIME TILL NEXT BREAK.
4346 023626 010046      MOV     RO,-(SP) ;SAVE CONTENTS OF RO FROM BREAK MACRO.
4347 023630      BREAK      ;CHECK FOR OPERATOR CONTROL/C.
4348 023632 012600      MOV     (SP)+,RO ;RESTORE CONTENTS OF RO.
4349 023634 000002 60$: RTI

```

```

4351 .SBTTL  INTERRUPT SERVICE ROUTINE          - RXBRRT -
4352 ;** *****
4353 ;*      - BR LEVEL TEST RECEIVE INTERRUPT SERVICE ROUTINE -
4354 ;*      THIS SERVICE ROUTINE HANDLES RECEIVE INTERRUPTS DURING THE INTERRUPT
4355 ;*      BR LEVEL TEST.  THIS ROUTINE COUNTS THE INTERRUPT AND SETS A FLAG
4356 ;*      TO INDICATE THAT THE INTERRUPT HAS OCCURRED.  IT ALSO CHECKS THE
4357 ;*      FLAG WHICH INDICATES THAT A TX INTERRUPT HAS OCCURRED.  IF THE TX
4358 ;*      INTERRUPT FLAG IS SET, THIS ROUTINE SETS AN INTERRUPT ORDER ERROR
4359 ;*      FLAG INDICATING THAT A TRANSMIT INTERRUPT WAS SERVICED BEFORE A
4360 ;*      SIMULTANEOUS RECEIVE INTERRUPT.
4361 ;*
4362 ;* INPUTS:      RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERUPTS.
4363 ;*              RXINTF - RX INTERRUPT FLAGS.
4364 ;*
4365 ;* OUTPUTS:     RXINTC - CONTAINS THE UPDATED INTERRUPT COUNT.
4366 ;*              RXINTF - RX INT FLAGS:
4367 ;*                  (BIT 0 SET, BIT 14 SET IF TXINTF BIT 0 IS SET.)
4368 ;*
4369 ;* CALLING SEQUENCE:  PUT THE ADDRESS OF THE LABEL RXBRRT IN THE VECTOR
4370 ;*                  LOCATION.
4371 ;*
4372 ;* COMMENTS:      NOTE: THE FIFO IS NOT PURGED BY THIS ROUTINE.
4373 ;*
4374 ;* SUBORDINATE ROUTINES CALLED: NONE.
4375 ;-- *****
4376
4377 023636 RXBRRT:: SAVE
4378 023636 004567 160130
4379 023642 017700 156372
4380 023646 016701 156436
4381 023652 005201
4382 023654 001402
4383 023656 010167 156426
4384 023662 016701 156424
4385 023666 052701 000001
4386 023672 032767 000001 156424
4387 023700 001402
4388 023702 052701 040000
4389 023706 010167 156400
4390 023712 004736
4391 023714 000002

```

```

                JSR      ;SAVE CONTENTS OF GPRS R0 THRU R5.
                MOV      R5,PREG05 ;CALL REGISTER SAVE SUBRT.
                MOV      BRBUFA,R0 ;READ THE CHAR OUT OF THE FIFO.
                MOV      RXINTC,R1 ;GET THE INTERRUPT COUNT.
                INL      R1         ;INCREMENT THE COUNT.
                BEQ      2$,        ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
                MOV      R1,RXINTC ;SAVE NEW COUNT VALUE.
2$:             MOV      RXINTF,R1 ;GET THE RX INTERRUPT FLAGS.
                BIS      #BIT0,R1  ;SET THE RX INTERRUPT HAS OCCURRED FLAG.
                BIT      #BIT0,TXINTF ;TEST THE "TX INT HAS OCCURRED" FLAG.
                BEQ      4$,        ;SKIP SLITTING ERROR FLAG IF NO TX INT.
                BIS      #BIT14,R1 ;SET THE INTERRUPT ORDER ERROR FLAG.
                MOV      R1,RXINTF ;UPDATE THE RX INTERRUPT FLAGS.
                PASS      60$,      ;RESTORE GPRS.
                JSR      PC,0(SP), ;RETURN TO PREG05 SUBRT.
                RTI

```

```

4392 .3BTTL  INTERRUPT SERVICE ROUTINE      - RXINPT -
4393 ;* *****
4394 ;* - RECEIVE CHARACTER INPUT INTERRUPT SERVICE ROUTINE -
4395 ;* THIS SERVICE ROUTINE INPUTS A CHARACTER FROM THE DUT AND LOADS THE
4396 ;* CHAR (COMPLETE WITH STATUS FLAGS) INTO A RECEIVE CHAR BUFFER IN
4397 ;* MEMORY. THE INTERRUPT IS ALSO COUNTED. THE RECEIVE CHAR BUFFER IS
4398 ;* MONITORED TO ENSURE THAT IT DOES NOT OVERFLOW.
4399 ;*
4400 ;* INPUTS:      BUFEND - LABELS THE END OF THE HOST MEMORY BUFFER.
4401 ;*              BUFPTR - CONTAINS ADDRESS OF NEXT FREE BUFFER LOCATION.
4402 ;*              CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
4403 ;*              RBUFA - CONTAINS THE ADDRESS OF THE RBUF DUT REGISTER.
4404 ;*              RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERRUPTS.
4405 ;*              RXINTF - RX INTERRUPT FLAGS.
4406 ;*
4407 ;* OUTPUTS:     BUFPTR - CONTAINS UPDATED ADDRESS OF NEXT FREE BUFFER LOCATION.
4408 ;*              RXINTC - CONTAINS THE UPDATED INTERRUPT COUNT.
4409 ;*              RXINTF - RX INT FLAGS (BIT 15 SET IF RX.DATA.AVAIL IS CLEAR).
4410 ;*
4411 ;* CALLING SEQUENCE:  PUT THE ADDRESS OF THE LABEL RXINPT IN THE VECTOR
4412 ;* LOCATION.
4413 ;*
4414 ;* COMMENTS:      IN CASE OF OVERFLOW OF THE MEMORY BUFFER, BUFPTR WILL BE
4415 ;*                  MAINTAINED EQUAL TO BUFEND AND THE WORD AT BUFPTR WILL BE
4416 ;*                  THE LAST WORD READ FROM THE DUT FIFO.
4417 ;*                  NOTE: THIS ROUTINE CAN DESTROY TX.ACTIONS BY READING THE CSR.
4418 ;*
4419 ;* SUBORDINATE ROUTINES CALLED: NONE.
4420 ;* -- *****
4421
4422 023716 RXINPT:: SAVE                                ;SAVE CONTENTS OF GPRS R0 THRU R5.
4423 023716 004567 160050                                R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4424 023722 017701 156310                                ;READ THE CONTENTS OF THE CSR.
4425 023732 001003 000200                                ;TEST RX.DATA.AVAIL BIT.
4426 023734 052767 100000 156350 2#: BNE 2#            ;BRANCH AROUND SETTING FLAG IF BIT IS SET.
4427 023742 016701 156342 2#: BIS #BIT15,RXINTF        ;SET THE RX.DATA.AVAIL CLEAR FLAG.
4428 023746 005201 156342 2#: MOV RXINTC,R1             ;GET THE INTERRUPT COUNT.
4429 023750 001402 156342 2#: INC R1                   ;INCREMENT THE COUNT.
4430 023752 010167 156332 2#: BEQ 4#                   ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
4431 023756 016702 156314 2#: MOV R1,RXINTC            ;SAVE NEW COUNT VALUE.
4432 023762 017722 156252 4#: MOV BUFPTR,R2            ;GET THE POINTER TO NEXT FREE BUFFER WORD.
4433 023766 020267 157730 4#: MOV RBUFA,(R2)+          ;READ A CHAR FROM THE FIFO INTO BUFFER.
4434 023772 103002 156276 4#: CMP R2,BUFEND            ;TEST FOR POINTER BEYOND END OF BUFFER.
4435 023774 010267 156276 4#: BHS 60#                 ;SKIP THE PTR UPDATE IF PTR OUT OF BOUNDS.
4436 024000 004736 156276 60#: MOV R2,BUFPTR          ;UPDATE THE BUFFER POINTER.
4437 024002 000002 156276 60#: PASS                    ;RESTORE GPRS.
4437 024002 000002 156276 60#: JSR PC,B(SP)+          ;RETURN TO PREG05 SUBRT.
4437 024002 000002 156276 60#: RTI

```



```
4469 .SBTTL INTERUPT SERVICE ROUTINE - TXINTR -
4470 ;* *****
4471 ;* - TRANSMIT INTERRUPT SERVICE ROUTINE -
4472 ;* THIS ROUTINE HANDLES A TRANSMIT INTERRUPT FROM THE DEVICE UNDER TEST
4473 ;* (DUT) BY COUNTING THE INTERRUPT AND READING THE DUT CSR TO CLEAR THE
4474 ;* INTERRUPT REQUEST. THIS ROUTINE ALSO SETS A FLAG TO INDICATE THAT
4475 ;* A TX INTERRUPT HAS OCCURRED AND SETS A FLAG IF THE TX.ACTION BIT IS
4476 ;* NOT SET IN THE READ CONTENTS OF THE DUT CSR.
4477 ;*
4478 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR.
4479 ;* TXINTC - HOLDS THE COUNT OF THE NUMBER OF TX INTERUPTS.
4480 ;* TXINTF - TX INTERRUPT FLAGS.
4481 ;*
4482 ;* OUTPUTS: TXINTC - CONTAINS THE UPDATED TX INTERRUPT COUNT.
4483 ;* TXINTF - TX INT FLAGS (BIT 0 SET, BIT 15 SET IF TX.ACTION CLR).
4484 ;*
4485 ;* CALLING SEQUENCE: PUT THE ADDRESS OF THE LABEL TXINTR IN THE VECTOR
4486 ;* LOCATION.
4487 ;*
4488 ;* COMMENTS:
4489 ;*
4490 ;* SUBORDINATE ROUTINES CALLED: NONE
4491 ;* --- *****
4492
4493 024026 TXINTR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
4494 024026 004567 157740 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4495 024032 016701 156264 ;GET THE TX INTERRUPT COUNT.
4496 024040 102001 ;INCREMENT THE COUNT.
4497 024042 005301 ;BRANCH IF NO OVERFLOW OCCURRED.
4498 024044 010167 156252 ;RESET THE COUNT TO 177777.
4499 024050 016703 156250 ;SAVE NEW COUNT VALUE.
4500 024054 017702 156156 ;GET THE TX INTERRUPT FLAGS.
4501 024060 100402 ;READ THE CSR.
4502 024062 052703 100000 ;SKIP SETTING OF FLAG IF TX.ACTION IS SET.
4503 024066 052703 000001 ;SET THE TX.ACTION CLEAR FLAG.
4504 024072 010367 156226 ;SET THE TX INT HAS OCCURRED FLAG.
4505 024076 ;UPDATE THE TX INTERRUPT FLAGS.
4506 024100 000002 ;RESTORE GPRS.
;PC,B(SP), ;RETURN TO PREG05 SUBRT.
RTI
```

```
4508
4509 ;*****
4510 ;
4511 ;          FVTA.RPT
4512 ;
4513 ;*****
4514
4515
4516
4517 .SBTTL  REPORT CODING SECTION
4518
4519 ;**
4520 ; THE REPORT CODING SECTION CONTAINS THE
4521 ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
4522 ;--
4523
4524 024102      BGNRPT
4525 024102
4526 024102      EXIT  RPT
4527 024102 000167
4528 024104 000000
4529
4530 024106
4531 024106
4532 024106 104425

                                L1RPT::
                                .WORD  J1JMP
                                .WORD  L10017-2-.

                                L10017:
                                TRAP    C1RPT
```

4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554

024110
024110

177777
177777
177777

.SBTTL PROTECTION TABLE

;
; FVTSKL4.P11
;

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

BGNPROT

L#PROT::

-1 ;OFFSET INTO P-TABLE FOR CSR ADDRESS
-1 ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
-1 ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

ENDPROT

```
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590 024116
      024116
4591
4592 024116 012700 000040
      024122 104447
4593 024124
      024124 103416
4594
4595 024126 012700 000037
      024132 104447
4596 024134
      024134 103556
4597
4598 024136 012700 000035
      024142 104447
4599 024144
      024144 103555
4600
4601 024146 012700 000036
      024152 104447
4602 024154
      024154 103161
4603 024156 000167 000540
4604 024162
4605 024162
      024162 104433
4606
4607
4608
4609 024164
      024164 012700 000114
      024170 104462

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

.SBTTL  INITIALIZE SECTION
;
;
;*****
;*   THIS SECTION CONTAINS THE CODE WHICH IS PERFORMED AT THE BEGINNING OF
;*   EACH PASS OR AFTER A CONTINUE COMMAND.
;*   THIS CODE PERFORMS THE FOLLOWING ACTIONS:
;*
;*   MOVES THE INFORMATION HELD IN THE HARDWARE P-TABLE INTO THE GLOBAL
;*   DATA AREA.
;*****
;--
      BGNINIT
;
;SEE IF PROGRAM JUST STARTED, BR IF YES
      READDEF  #EF.START
;
;*****
;*   BCOMPLETE      NEWSTA
;
;SEE IF PROGRAM JUST RESTARTED, BR IF YES
      READDEF  #EF.RESTART
;
;*****
;*   BCOMPLETE      NEWRES
;
;SEE IF THIS IS A NEW PASS, BR IF YES
      READDEF  #EF.NEW
;
;*****
;*   BCOMPLETE      NEWPAS
;
;SEE IF PROGRAM WAS JUST CONTINUED
      READDEF  #EF.CONTINUE
;
;*****
;*   BNCOMPLETE     GETPRM
;
;*****
;*   JMP            ENDIT
;
;*****
;*   NEWSTA:
;*   BRESET
;*   ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
;*   TRAP          C$RESET
;
;*****
;*   SET UP FOR LINE TIME CLOCK INTERRUPTS.
;
;*****
;*   CLOCK  L,R1
;*   ;GET THE CLOCK PARAMETERS.
;
;*****
;*   MOV
;*   TRAP          #'L,RO
;*               C$CLCK
```

```

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

```

```
4658 024434 005067 155720          CLR    MMENAB          ;INDICATE MEM MGT IS NOT ENABLED.
4659 024440 004767 173242          JSR    PC,CKTRAP        ;CLEAR THE MEM MGT SRO REG AND CHECK FOR TRAP.
4660 024444 016767 155646 153332    MOV    TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4661 024452 103003                    BCC    10$              ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
4662 024454 012767 000001 155674    MOV    #1,MMPRES        ;INDICATE THAT MEM MGT IS PRESENT.
4663 024462 005067 155620 10$:     CLR    PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4664 024466 000167 000006          JMP    NEWPAS          ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
4665
4666 024472          NEWRES: BRESET          ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
4667 024472 104433                    CLR    PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4668 024474 005067 155606          NEWPAS:
4669 024500 012767 177777 155524    MOV    #-1,UNITN        ;RESET LOGICAL DEVICE TO -1
4670
4671          ;* INCREMENT THE PASS COUNTER, CORRECT FOR ANY OVERFLOW.
4672          ; THIS COUNTER IS USED IN THE ROM VERSION TEST.
4673          ;-
4674 024506 005267 155574          INC    PASCNT          ;INCREMENT THE PASS COUNTER.
4675 024512 001002                    BNE    GETPRM          ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
4676 024514 005367 155566          DEC    PASCNT          ;SET PASS COUNT TO 177777 OCTAL.
4677
4678          ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
4679 024520          GETPRM:
4680 024520 005267 155506          INC    UNITN          ;INCREMENT LOGICAL DEVICE NUMBER
4681 024524 026767 155502 155260    CMP    UNITN,L$UNIT        ;SEE IF MAXIMUM UNIT NO. EXCEEDED
4682 024532 002362                    BGE    NEWPAS          ;BR IF YES
4683
4684 024534          GPHARD UNITN,R1          ;GET P-TABLE POINTER INTO R1
4685 024534 016700 155472          MOV    UNITN,R0          ;MOV UNITN,R0
4686 024540 104442          TRAP    C$GPHRD          ;TRAP C$GPHRD
4687 024542 010001          MOV    R0,R1          ;MOV R0,R1
4688
4689          BCOMPLETE 30$              ;BR IF DEVICE AVAILABLE
4690 024544 103401          BR    GETPRM          ;SKIP THIS DEVICE
4691 024546 000764          BCS    30$
4692
4693          ;***** HARDWARE PARAMETER MOVING CODE *****
4694 024550 012167 155462 30$:     MOV    (R1)+,CSRA        ;STORE DHU-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
4695 024554 012102          MOV    (R1)+,R2          ;GET THE RX INTERRUPT VECTOR ADDRESS.
4696 024556 010267 155444          MOV    R2,RXVECA        ;STORE RX INT VECTOR ADDRESS.
4697 024562 062702 000004          ADD    #4,R2          ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
4698 024566 010267 155436          MOV    R2,TXVECA        ;STORE TX INT VECTOR ADDRESS.
4699 024572 012167 155426          MOV    (R1)+,ACTLNS        ;STORE DHU-11 ACTIVE LINE BIT MAP
4700 024576 111167 155432          MOVB   (R1),BRLEVL        ;STORE DHU-11 INTERRUPT BUS REQUEST LEVEL
4701
4702          ;*
4703          ; CALCULATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
4704          ; DEVICE REGISTER ADDRESS TABLE.
4705          ;-
4706 024602 016701 155430          MOV    CSRA,R1          ;COPY CSR ADDRESS
4707 024606 005201          INC    R1          ;INCREMENT CSR ADDRESS
4708 024610 005201          INC    R1          ; COPY BY 2.
4709 024612 012703 000007          MOV    #7,R3          ;SET UP REGISTER COUNT
4710 024616 012702 002240          MOV    #RBUFA,R2        ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
4711 024622 010122 12$:     MOV    R1,(R2)+          ;STORE REGISTER ADDRESS IN TABLE
4712 024624 005201          INC    R1          ;INCREMENT REGISTER ADDRESS
4713 024626 005201          INC    R1          ; BY 2,FOR THE NEXT DEVICE REGISTER.
4714 024630 005303          DEC    R3          ;DECREMENT REGISTER COUNT
```

```

4710 024632 001373          BNE      12$          ;LOOP IF NOT DONE
4711
4712
4713          ;*
4714          ; INITIALISE THE BMP CODE QUEUE.
4715 024634 012700 002522          MOV      #BMPQ8,R0          ;GET THE START ADDRESS OF THE QUEUE.
4716 024640 012701 002722          MOV      #BMPQ8,R1          ;GET THE END ADDRESS OF THE QUEUE.
4717 024644 010067 155650          MOV      R0,BMPQ8          ;SET THE POINTER TO THE START OF THE QUEUE.
4718 024650 005020          14$: CLR      (R0),          ;CLEAR OUT THE CONTENTS OF THE QUEUE.
4719 024652 020001          CMP      R0,R1          ;CHECK IF END OF QUEUE HAS BEEN REACHED.
4720 024654 103775          BLO      14$          ;LOOP IF NOT ALL DONE.
4721
4722          ;*
4723          ; REPORT THE UNIT NUMBER IF THE SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
4724          ; AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
4725 024656 032767 000020 155334          BIT      #BIT4,OPTION          ;CHECK IF THE QUESTION WAS ANSWERED YES.
4726 024664 001416          BEQ      16$          ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
4727 024666 026727 155120 000001          CMP      L$UNIT,#1          ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
4728 024674 003412          BLE      16$          ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
4729 024676          PRINTF      #MFUNIT,UNITN          ;REPORT UNIT NUMBER.
4730 024676 016746 155330          MOV      UNITN,-(SP)
4731 024702 012746 004070          MOV      #MFUNIT,-(SP)
4732 024706 012746 000002          MOV      #2,-(SP)
4733 024712 010600          MOV      SP,R0
4734 024714 104417          TRAP      C$PNTF
4735 024716 062706 000006          ADD      #6,SP
4736 024722          16$:
4737 024722 005067 155354          ENDIT: CLR      CTRLCF          ;CLR THE CTRL-C TEST ABORT FLAG.
4738          ;*
4739          ; SET THE PROCESSOR PRIORITY TO ALLOW LTC INTERRUPTS BUT NOT OTHERS.
4740          ;*
4741          SETPRI      #PRI07          ;SET PROCESSOR PRIORITY TO 7.
4742 024726 012700 000340          MOV      #PRI07,R0
4743 024732 104441          TRAP      C$SPRI
4744 024734          ENDINIT
4745 024734 104411          L10021: TRAP      C$INIT
4746 000000          TNUM == 0          ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.

```

```

4742 ;*****
4743 ;
4744 ;           FVTA.ATD
4745 ;
4746 ;*****
4748
4749
4750 .SBTTL AUTODROP SECTION
4751
4752
4753 ;**
4754 ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
4755 ; THE "ADR" FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
4756 ; SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
4757 ; DROPPED FROM TESTING.
4758 ;--
4759
4760 024736          BGNAUTO
4761 024736
4762
4763
4764
4765
4766
4767
4768
4769 024736          ENDAUTO
4770 024736
4771 024736 104461
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```

```

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

.SBTTL  CLEANUP CODING SECTION

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

      BGNCLN

                                L$CLEAN::

      TST      CTRLCF           ;DID WE GET HERE BY CTRL-C FROM TEST?
      BEQ      2$              ;CTRL-C FROM TEST? NO, SKIP BUS RESET.
      BRESET   ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
                                TRAP      C$RESET

2$:
      EXIT     CLN

                                TRAP      C$EXIT
                                .WORD     L10023-.

      .EVEN

      ENDCLN

                                L10023:
                                TRAP      C$CLEAN

```

```
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835 024756
      024756
4836
4837
4838
4839
4840
4841
4842
4843
4844 024756
      024756 010046
      024760 012746 025002
      024764 012746 000002
      024770 010600
      024772 104417
      024774 062706 000006
4845 025000 000427
4846
4847 025002 045 101 040 DROP: .ASCIZ/ A UNIT D6 A DROPPED FROM FURTHER TESTING. N/
      025005 125 116 111
      025010 124 045 104
      025013 066 045 101
      025016 040 104 122
      025021 117 120 120
      025024 105 104 040
      025027 106 122 117
      025032 115 040 106
      025035 125 122 124
      025040 110 105 122
      025043 040 124 105
      025046 123 124 111
      025051 116 107 056
      025054 045 116 000
4848
4849 025060
4850 025060
      025060 000167
      025062 000000
4851
4852
```

```
*****
;
;          FVTA.DRP
;
*****

.SBTTL  DROP UNIT SECTION

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

      BGNDU

                                  L#DU::
*****
;
;   INSERT DROP CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
;   A "DROP" COMMAND OR A "DODU" MACRO EXECUTION.  THE PURPOSE
;   OF THIS CODE IS TO DO ANY NECESSARY HOUSEKEEPING AFTER A
;   UNIT HAS BEEN DROPPED.  THIS SECTION IS OPTIONAL.
*****

      PRINTF #DROP,R0          ;REPORT UNIT THAT HAS BEEN DROPPED.
                                  MOV      R0,-(SP)
                                  MOV      #DROP,-(SP)
                                  MOV      #2,-(SP)
                                  MOV      SP,R0
                                  TRAP     C#PNTF
                                  ADD      #6,SP

      BR      EDROP          ;BRANCH AROUND THE MESSAGE.

      DROP:  .ASCIZ/ A UNIT D6 A DROPPED FROM FURTHER TESTING. N/

      .EVEN
EDROP:  EXIT      DU

      .WORD   J#JMP
      .WORD   L10024 2-.
```

M10

CZDHUAO DHU-11 FUNC TST PART1 MACRO M1200 12 DEC 83 16:08 PAGE 95 1
DROP UNIT SECTION

SEQ 129

4853
4854 025064
025064
025064 104453

ENDDU

L10024: TRAP C#DU

```

4856
4857 ;*****
4858 ;
4859 ;           FVTA.ADD
4860 ;
4861 ;*****
4862
4863
4864
4865 .SBTTL  ADD UNIT SECTION
4866
4867 ;++
4868 ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
4869 ; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
4870 ; TO THE TEST CYCLE.
4871 ;--
4872
4873 025066          BGNAU
4874                L$AU::
4875
4876 ;*****
4877 ;           INSERT ADD CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
4878 ;           AN "ADD" COMMAND.  THE PURPOSE OF THIS CODE IS TO DO ANY
4879 ;           HOUSEKEEPING THAT MAY BE NECESSARY AFTER A UNIT HAS BEEN ADDED.
4880 ;           THIS SECTION IS OPTIONAL.
4881 ;*****
4882 025066          EXIT    AU
4883 025066 000167
4884 025070 000000
4885
4886
4887 025072          .EVEN
4888 025072          ENDAU
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```

.WORD J\$JMP
 .WORD L10025-2-.

L10025: TRAP C\$AU

```

TNUM == TNUM + 1      ; INCREMENT THE ASSEMBLY TIME TEST COUNTER.
MOV     @TNUM,TSTNUM   ; SET UP THE TEST NUMBER. (1)
MOV     @-1,CTRLCF     ; INDICATE THAT WE ARE IN A TEST.
MOV     @101.,ERRNBR   ; SET THE TEST ERROR NUMBER IN THE TABLE.
MOV     @EM0103,ERRMSG ; SET UP THE TEST FAILURE MESSAGE IN THE TABLE.
MOV     @ERO101,ERRBLK ; SET-UP THE ERROR ROUTINE IN THE ERROR TABLE.

; SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
;
MOV     4,TP4VEC       ; SAVE THE EXISTING 004 TRAP VECTOR.
MOV     @TP4RTN,4      ; SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
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.
;
MOV     CSRA,R0        ; SET UP CSR AS THE CKTRAP MOVE SOURCE.
MOV     @520,R1        ; SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
JSR     PC,CKTRAP      ; MOVE AND CHECK FOR TRAP.
BCS     40             ; IF NO TRAP, BYPASS ERROR.
BIS     @100001,R5     ; SET FATAL READ ERROR FLAGS.
40:     BIC     @17,520  ; CLEAR THE I.A.R FIELD OF THE CSR DATA.
MOV     R1,R0          ; USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
MOV     CSRA,R1        ; SET UP CSR AS THE CKTRAP MOVE DESTINATION.
JSR     PC,CKTRAP      ; MOVE AND CHECK FOR TRAP.
BCS     60             ; IF NO TRAP, BYPASS ERROR.
BIS     @100002,R5     ; SET FATAL WRITE ERROR FLAGS.
BR      400            ; EXIT AND REPORT FATAL ERROR.

; NOW, WE TEST EACH REGISTER FOR THIS LINE.
;
60:     MOV     @8.,R2   ; INIT REGISTER COUNTER TO 8.
MOV     CSRA,500       ; INITIALIZE THE REGISTER POINTER.
80:     MOV     @500,R0  ; SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
MOV     @520,R1        ; SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
JSR     PC,CKTRAP      ; PERFORM THE MOVE, CHECK FOR TRAP.
BCS     100            ; IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
BIS     @100001,R5     ; SET FATAL READ ERROR FLAGS.
100:    MOV     R1,R0    ; USE OLD DEST AS SRC FOR CKTRAP MOVE.
MOV     @500,R1        ; SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.

```

C11

```

4946 025264 004767 172416      JSR      PC,CKTRAP      ;PERFORM THE MOVE, CHECK FOR TRAP.
4947 025270 103402              BCS      12$      ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
4948 025272 052705 100002      BIS      #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
4949 025276 005267 000042      12$: INC      50$      ;INCREMENT THE REGISTER
4950 025302 005267 000036      INC      50$      ; POINTER BY 2.
4951 025306 005302              DEC      R2      ;COUNT THE REGISTER.
4952 025310 001351      BNE      8$      ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
4953
4954
4955      ;*
4956      ; DONE CHECKING DEVICE REGISTER ADDRESSES.
4957      ; REPORT ANY ERRORS AND EXIT.
4958      ;*
4959 025312 016767 155000 152464 40$: MOV      TP4VEC,4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
4960 025320 005705              TST      R5      ;CHECK THE ERROR FLAGS.
4961 025322 100012      BPL      60$      ;EXIT ROUTINE IF NO ERRORS.
4962
4963      ;*
4964      ; REPORT "DEVICE REGISTER ACCESS TEST FAILED"
4965      ;*
4965 025324              ERROR
4966 025324 104460
4967
4968      DODU      UNITN      ;DROP THIS UNIT FROM FUTHER TESTING.
4969 025326 016700 154700      MOV      UNITN,R0
4970 025332 104451      TRAP      C#DODU
4971 025334 005067 154742      CLR      CTRLCF      ;INDICATE NO CTRL-C ABORT FROM TEST.
4972 025340 104444      DOCLN      ;ABORT THIS SUB PASS.
4973 025342 000402      BR      60$      TRAP      C#DCLN
4974
4975      ;***** LOCAL STORAGE. *****
4976 50$: .WORD      0      ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
4977 52$: .WORD      0      ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
4978      ;***** END *****
4979 60$: CLR      CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
4980      ENDTST
4981
4982      L10026:
4983      TRAP      C#ETST

```

```

4981 .SBTTL  HARDWARE TEST          - MRSTA -
4982 ;* *****
4983 ;* - MASTER RESET WITH SELFTEST TEST -
4984 ;* THIS TEST VERIFIES THAT THE MASTER RESET BIT WILL CLEAR AFTER A DEVICE
4985 ;* RESET AND THE PERFORMANCE OF THE DUT ROM BASED SELFTEST.
4986 ;*
4987 ;* *****
4988 025356 BGNTST
4989 025356 000002 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
4990 025356 012767 000002 154734 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (2)
4991 025364 012767 177777 154710 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
4992 025372 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
4993 025400 012700 000240 MOV #PRI05,R0
4994 025406 012767 000001 156354 TRAP C#SPRI
4995 025414 012767 005632 156352 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
4996 025414 012767 015776 156346 MOV #EM0201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
4997 ;*
4998 ;* WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
4999 025422 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5000 025426 012702 000040 MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5001 025432 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5002 025434 016704 154576 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5003 025440 004767 172454 JSR PC,MSLGET ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5004 025444 103410 BCS 2# ;SKIP TO RESET DUT IF MR CLEAR.
5005 ;*
5006 ;* DUT MASTER RESET BIT DID NOT GO CLEAR. DEVICE MAY BE STUCK IN SOME
5007 ;* ODD STATE. TRY TO RESET DEVICE WITH A BUS RESET.
5008 ;*
5009 025446 BRESET ;NO. TRY TO JOG DEVICE WITH BUS RESET.
5010 025446 104433 TRAP C#RESET
5011 025450 004767 174624 JSR PC,SKPSTS ;TRY TO SKIP THE SELFTEST.
5012 025454 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5013 025464 004767 172434 JSR PC,MSLGET ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5014 025464 103016 BCC 4# ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
5015 ;*
5016 ;* SET THE MASTER RESET BIT AND VERIFY THAT IT CLEARS WITHIN THE PROPER TIME.
5017 025466 012701 011610 2# : MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5018 025472 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5019 025474 004767 172420 JSR PC,MSLGET ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5020 025500 103010 BCC 4# ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
5021 025502 012702 011610 MOV #5000.,R2
5022 025506 160102 SUB R1,R2 ;CALCULATE # OF MS FOR MR TO CLEAR.
5023 025510 001413 BEQ 6# ;GO REPORT ERROR IF MR CLEAR IMMEDIATELY.
5024 025512 020227 000764 CMP R2,#500.
5025 025516 002417 BLT 8# ;GO REPORT ERROR IF MR CLEAR IN < 1/2 SECOND.
5026 025520 000424 BR 60# ;EXIT THE TEST WITHOUT ERROR.
5027 ;*
5028 ;* ERROR REPORTS:
5029 ;*
5030 ;* REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5031 025522 012767 000311 156234 4# : MOV #201.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5032 025530 012701 005665 MOV #EM0202,R1 ;SELECT ERROR MESSAGE.
5033 025534 ERROR ;REPORT ERROR. >>>> ERROR #201 <<<<

```

```

025534 104460
5034 025536 000415 BR 60$ ;EXIT THE TEST. TRAP C$ERROR
5035
5036
5037 025540 012767 000312 156216 6$: ;REPORT MR BIT CLEAR IMMEDIATELY AFTER DUT RESET.
5038 025546 012701 006040 MOV #202,,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5039 025552 MOV #EM0203,R1 ;SELECT ERROR MESSAGE.
025552 104460 ;REPORT ERROR. >>>> ERROR #202 <<<<
5040 025554 000406 BR 60$ ;EXIT THE TEST. TRAP C$ERROR
5041
5042
5043 025556 012767 000313 156200 8$: ;REPORT MR CLEAR WITHIN 1/2 SECOND OF DUT RESET.
5044 025564 012701 006203 MOV #203,,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5045 025570 MOV #EM0204,R1 ;SELECT ERROR MESSAGE.
025570 104460 ;REPORT ERROR. >>>> ERROR #203 <<<<
5046 TRAP C$ERROR
5047 025572 60$: SETPRI #PRI07 ;DISABLE ALL INTERRUPTS.
025572 012700 000340 MOV #PRI07,R0
025576 104441 TRAP C$SPRI
5048 025600 005067 154476 CLR CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
5049 025604 ENDTST
025604 104401 L10027: TRAP C$ETST

```

```
5051 .SBTTL  HARDWARE TEST          - MRSSTA -
5052 ;* *****
5053 ;* - MASTER RESET WITH SKIP SELFTEST TEST -
5054 ;* THIS TEST VERIFIES THAT THE MASTER RESET BIT WILL CLEAR AFTER A DEVICE
5055 ;* RESET AND THE SKIPPING OF THE DUT ROM BASED SELFTEST.
5056 ;*
5057 ;* *****
5058 025606 BGNTST
5059 025606 000003 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5060 025606 012767 000003 154504 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (3)
5061 025614 012767 177777 154460 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
5062 025622 025622 012700 000240 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
5063 025630 012767 000001 156124 MOV #1,ERRTP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5064 025636 012767 006362 156122 MOV #EM0301,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5065 025644 012767 015776 156116 MOV #ER0201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5066 ;*
5067 ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5068 ;*
5069 025652 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5070 025656 012702 000040 MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5071 025662 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5072 025664 016704 154346 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5073 025670 004767 172224 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5074 025674 103410 BCS 2# ;SKIP TO RESET DUT IF MR CLEAR.
5075 ;*
5076 ; DUT MASTER RESET BIT DID NOT GO CLEAR. DEVICE MAY BE STUCK IN SOME
5077 ; ODD STATE. TRY TO RESET DEVICE WITH A BUS RESET.
5078 ;*
5079 025676 025676 104433 BRESET ;NO. TRY TO JOG DEVICE WITH BUS RESET.
5080 025700 004767 174374 JSR PC,SKPSTS ;TRY TO SKIP THE SELFTEST.
5081 025704 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5082 025710 004767 172204 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5083 025714 103024 BCC 6# ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
5084 ;*
5085 ; SET THE MASTER RESET BIT, TRY TO SKIP THE SELFTEST, AND VERIFY THAT THE
5086 ; MR BIT CLEARS WITHIN 1/5 SECOND.
5087 ;*
5088 025716 012701 000310 2# MOV #200.,R1 ;TIME-OUT VALUE IS 1/5 SECOND.
5089 025722 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5090 025724 004767 174350 JSR PC,SKPSTS ;TRY TO SKIP THE SELFTEST.
5091 025730 004767 172164 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5092 025734 103007 BCC 4# ;GO FIND OUT WHAT IS WRONG IF MR NOT CLEAR.
5093 025736 012702 000310 MOV #200.,R2
5094 025742 160102 SUB R1,R2 ;CALCULATE # OF MS FOR MR TO CLEAR.
5095 025744 020227 000012 CMP R2,#10.
5096 025750 002415 BLT 8# ;GO REPORT ERROR IF MR CLEAR IN < 10 MS.
5097 025752 000431 BR 60# ;EXIT THE TEST WITHOUT ERROR.
5098 ;*
5099 ; MR DID NOT CLEAR WITHIN 1/5 SECOND, SEE IF IT CLEARS WITHIN 5 SECONDS.
5100 ;*
5101 025754 012701 011300 4# MOV #4800.,R1 ;TIME-OUT VALUE IS 5 SECONDS MINUS 1/5 SECOND.
5102 025760 004767 172134 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5103 025764 103416 BCS 10# ;GO REPORT ERROR IF MR CLEARED FINALLY.
```

```

5104
5105      ;
5106      ; ERROR REPORTS:
5107      ;
5108      ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5109      025766 012767 000455 155770 6+:      MOV    #0301.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5110      025774 012701 005665      MOV    #EM0202,R1      ;SELECT ERROR MESSAGE.
5111      026000      104460      ERROR      ;REPORT ERROR.          >>>>> ERROR #0301 <<<<<
5112      026000      104460      BR        60+      ;EXIT THE TEST.          TRAP    C#ERROR
5113
5114      ;REPORT MR BIT CLEAR WITHIN 10 MS AFTER DUT RESET.
5115      026004 012767 000456 155752 8+:      MOV    #0302.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5116      026012 012701 006434      MOV    #EM0302,R1      ;SELECT ERROR MESSAGE.
5117      026016      104460      ERROR      ;REPORT ERROR.          >>>>> ERROR #0302 <<<<<
5118      026020 000406      BR        60+      ;EXIT THE TEST.          TRAP    C#ERROR
5119
5120      ;REPORT MR CLEARED BETWEEN 1/5 SECOND AND 5 SECONDS OF DUT RESET.
5121      026022 012767 000457 155734 10+:     MOV    #0303.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5122      026030 012701 006574      MOV    #EM0303,R1      ;SELECT ERROR MESSAGE.
5123      026034      104460      ERROR      ;REPORT ERROR.          >>>>> ERROR #0303 <<<<<
5124      026036      104460      BR        60+      ;EXIT THE TEST.          TRAP    C#ERROR
5125      026042 012700 000340      60+:      SETPRI  #PRI07      ;DISABLE ALL INTERRUPTS.
5126      026044 005067 154232      CLR     CTRLCF      ;INDICATE THAT WE COMPLETED THE TEST.
5127      026050      104401      ENDTST
5128      026050      104401      L10030:
5129      026050      104401      TRAP    C#ETST

```

H11

```
5128 .SBTTL HARDWARE TEST - RXCHRA -
5129 ;** *****
5130 ;* - RBUF REGISTER RX CHARACTER FIELD TEST -
5131 ;* THIS TEST VERIFIES THAT THE RX CHARACTER FIELD OF THE DUT RBUF REGISTER
5132 ;* APPEARS TO BE FUNCTIONING CORRECTLY. THIS TEST USES THE CODES WHICH
5133 ;* SHOULD BE IN THE FIFO AFTER A BOARD RESET AND SKIP SELFTEST SEQUENCE.
5134 ;*
5135 ;-- *****
5136 026052 BGNST
5137 026052 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T4::
5138 026052 012700 000240 MOV #PRI05,R0
5139 026052 104441 TRAP C#SPRI
5138 000004 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5139 026060 012767 000004 154232 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (4)
5140 026066 012767 177777 154206 MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5141 026074 012767 000001 155660 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5142 026102 012767 006753 155656 MOV #EM0401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5143 026110 012767 015776 155652 MOV #ER0201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5144
5145 ;* SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5146 ;* AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5147 ;--
5148 026116 012701 011610 MOV #5000,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5149 026122 012702 000040 MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5150 026126 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5151 026130 016704 154102 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5152 026134 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5153 026136 004767 174136 JSR PC,SKPSTS ;SKIP THE SELFTEST.
5154 026142 004767 171752 JSR PC,MSLGET ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5155 026146 103015 BCC 4# ;GO REPORT ERROR IF MR DID NOT CLEAR.
5156
5157 ;* READ 6 CHARACTERS FROM THE DUT AND VERIFY THAT THEY ARE VALID SELFTEST
5158 ;* CODES.
5159 ;--
5160 026150 012400 MOV (R4)+,R0 ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
5161 026152 012701 000006 MOV #6,R1 ;INITIALIZE THE LOOP COUNTER.
5162 026156 011402 2# MOV (R4),R2 ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
5163 026160 010200 MOV R2,R0
5164 026162 042700 177476 BIC #177476,R0 ;REMOVE ALL BUT BITS SPECIFIC TO SELFTEST CODE.
5165 026166 020027 000201 CMP R0,#201 ;CHECK THAT BITS 0,6, AND 7 ARE CORRECT.
5166 026172 001012 BNE 6# ;GO REPORT ERROR IF CODE IS NOT SELFTEST CODE.
5167 026174 005301 DEC R1 ;COUNT THIS LOOP ITERATION.
5168 026176 001367 BNE 2# ;LOOP IF NOT ALL LINES DONE.
5169 026200 000415 BR 60# ;EXIT TEST, NO ERROR FOUND.
5170
5171 ;* ERROR REPORTS:
5172 ;--
5173
5174 ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5175 026202 012767 000621 155554 4# MOV #0401,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5176 026210 012701 005665 MOV #EM0202,R1 ;SELECT ERROR MESSAGE.
5177 026214 ERROR ;REPORT ERROR. >>>>> ERROR #0401 <<<<<
5178 026214 104460 TRAP C#ERROR
5179 026216 000406 BR 60# ;EXIT THE TEST.
5180 ;REPORT IMPROPER CODE FOUND IN DUT RBUF AFTER RESET (SKIP SELFTEST).
```

```

5181 026220 012767 000622 155536 61:  MOV  #0402.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5182 026226 012701 007031  MOV  #EM0402,R1 ;SELECT ERROR MESSAGE.
5183 026232  ERROR  ;REPORT ERROR. >>>>> ERROR #0402 <<<<<
      026232 104460  TRAP  C#ERROR
5184
5185 026234 601:  SETPRI #PRI07 ;DISABLE ALL INTERRUPTS.
      026234 012700 000340  MOV  #PRI07,R0
      026240 104441  TRAP  C#SPRI
5186 026242 005067 154034  CLR  CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
5187 026246  ENDTST
      026246 104401  L10031:
      J26246  TRAP  C#ETST

```

```

5189 .SBTTL  HARDWARE TEST          - RXFFDA -
5190 ;+ *****
5191 ;+ - RBUF REGISTER RX FLAG FIELD TEST -
5192 ;+ THIS TEST VERIFIES THAT THE FIELD OF 3 FLAG BITS IN THE RBUF READS
5193 ;+ AS ALL ONES WHEN THE SELFTEST CODES ARE BEING READ FROM THE DUT
5194 ;+ AFTER A BOARD RESET AND SKIP SELFTEST SEQUENCE.
5195 ;+
5196 ;-- *****
5197 026250 BGNTST
5198 026250          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.          T5::
5199 026250          MOV      #PRI05,R0
5200 026254          TRAP    C#SPRI
5201 026256 012700 000005 154034      TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5202 026264 012767 000005 154010      MOV      #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (5)
5203 026272 012767 177777 154010      MOV      #-1,CTRLCF   ;INDICATE THAT WE ARE WITHIN A TEST.
5204 026300 012767 000001 155462      MOV      #1,ERRTYP    ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5205 026306 012767 007201 155460      MOV      #EM0501,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5206 026306 012767 015776 155454      MOV      #ERO201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5207 ;+
5208 ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5209 ; AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5210 ;--
5211 026314 012701 011610          MOV      #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
5212 026320 012702 000040          MOV      #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
5213 026324 005003          CLR      R3                    ;WAITING FOR BIT TO CLEAR.
5214 026326 016704 153704          MOV      CSRA,R4         ;BIT IS IN THE DUT'S CSR.
5215 026332 010214          MOV      R2,(R4)              ;SET THE DUT MASTER RESET BIT.
5216 026334 004767 173740          JSR      PC,SKPSTS       ;SKIP THE SELFTEST.
5217 026340 004767 171554          JSR      PC,MSLGET       ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5218 026344 103013          BCC      4#                    ;GO REPORT ERROR IF MR DID NOT CLEAR.
5219 ;+
5220 ; READ 8 CHARACTERS FROM THE DUT AND VERIFY THAT ALL 3 RX ERROR FLAGS ARE
5221 ; SET FOR EACH CHARACTERS.
5222 ;--
5223 026346 012400          MOV      (R4)+,R0              ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
5224 026350 012701 000010          MOV      #8.,R1         ;INITIALIZE THE LOOP COUNTER.
5225 026354 011402          MOV      (R4),R2              ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
5226 026356 012700 070000          MOV      #70000,R0
5227 026362 040200          BIC      R2,R0                ;CALCULATE BIT MAP OF CLEAR RX ERROR FLAGS.
5228 026364 001012          BNE      6#                    ;GO REPORT ERROR IF NOT ALL RX ERROR FLAGS SET.
5229 026366 005301          DEC      R1                    ;COUNT THIS LOOP ITERATION.
5230 026370 001371          BNE      2#                    ;LOOP IF NOT ALL LINES DONE.
5231 026372 000415          BR       60#                   ;EXIT TEST, NO ERROR FOUND.
5232 ;+
5233 ; ERROR REPORTS:
5234 ;--
5235 026374 012767 000765 155362 4# : ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5236 026402 012701 005665          MOV      #0501.,ERRNBR   ;SET THE ERROR NUMBER IN ERROR TABLE.
5237 026406          MOV      #EM0202,R1                    ;SELECT ERROR MESSAGE.
5238 026406 104460          ERROR                                ;REPORT ERROR. >>>> ERROR #0501 <<<<
5239 026410 000406          BR       60#                    ;EXIT THE TEST.          TRAP    C#ERROR
5240 ;+
5241 026412 012767 000766 155344 6# : ;REPORT ONE OR MORE RX ERROR FLAGS FOUND CLEAR WITH SELFTEST CODE.
5242          MOV      #0502.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.

```

[illegible]

L11

```
5249 .SBTTL HARDWARE TEST - RDAA -
5250 ;* *****
5251 ;* - CSR RX DATA AVAILABLE BIT TEST -
5252 ;* THIS TEST VERIFIES THAT THE DUT CSR RX DATA AVAILABLE BIT IS SET BY THE
5253 ;* INCLUSION OF THE SELFTEST CODES IN THE DUT FIFO AND THAT THE BIT CLEARS
5254 ;* AFTER THE FIFO HAS BEEN EMPTIED.
5255 ;*
5256 ;* *****
5257 ;* BGNTST
5258 026442 012700 000240 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T6::
5259 026442 104441 MOV #PRI05,R0
5260 026446 000006 TRAP C#SPRI
5261 026450 012767 000006 153642 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5262 026456 012767 177777 153616 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (6)
5263 026464 012767 000001 155270 MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5264 026472 012767 007611 155266 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5265 026500 012767 015776 155262 MOV #EM0601,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5266 ;*
5267 ;* SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5268 ;* AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5269 026506 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5270 026512 012702 000040 MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5271 026516 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5272 026520 016704 153512 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5273 026524 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5274 026526 004767 173546 JSR PC,SKPSTS ;SKIP THE SELFTEST.
5275 026532 004767 171362 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5276 026536 103016 BCC 4# ;GO REPORT ERROR IF MR DID NOT CLEAR.
5277 ;*
5278 ;* CHECK THAT THE RX DATA AVAILABLE BIT IS SET.
5279 ;*
5280 026540 032714 000200 BIT #BIT7,(R4) ;TEST THE DUT RX.DATA.AVAIL BIT.
5281 026544 001422 BEQ 6# ;GO REPORT ERROR IF BIT IS NOT SET.
5282 ;*
5283 ;* READ CHARACTERS FROM THE DUT RX FIFO AND WAIT FOR RX.DATA.AVAIL TO GO CLEAR.
5284 ;*
5285 026546 012705 001130 MOV #600.,R5 ;ALLOW READING 600 CHARS BEFORE ERROR.
5286 026552 010403 MOV R4,R3
5287 026554 012300 MOV (R3)+,R0 ;CALCULATE THE RBUF ADDRESS.
5288 026556 011300 MOV (R3),R0 ;READ A CHARACTER FROM THE RX FIFO.
5289 026560 032714 000200 BIT #BIT7,(R4) ;TEST THE DUT RX.DATA.AVAIL BIT.
5290 026564 001427 BEQ 60# ;EXIT TEST WITHOUT ERROR IF RX.DATA.AVAIL CLR.
5291 026566 005305 DEC R5 ;COUNT THE CHARACTER JUST READ.
5292 026570 001372 BNE 2# ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
5293 026572 000416 BR 8# ;GO REPORT ERROR IF RX.DATA.AVAIL WOULDN'T CLR.
5294 ;*
5295 ;* ERROR REPORTS:
5296 ;*
5297 ;*
5298 ;* REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5299 026574 012767 001131 155162 4# MOV #0601.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5300 026602 012701 005665 MOV #EM0202,R1 ;SELECT ERROR MESSAGE.
5301 026606 104460 ERROR ;REPORT ERROR. >>>> ERROR #0601 <<<<
; TRAP C#ERROR
```

```

5302 026610 000415          BR      60$          ;EXIT THE TEST.
5303
5304          ;REPORT THAT RX.DATA.AVAIL BIT WAS NOT SET AFTER A RESET COMPLETION.
5305 026612 012767 001132 155144 6$:      MOV      #C602.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5306 026620 012701 007653          MOV      #EM0602,R1      ;SELECT ERROR MESSAGE.
5307 026624          ERROR          ;REPORT ERROR.          >>>>> ERROR #0602 <<<<<
          026624 104460          TRAP      C#ERROR
5308 026626 000406          BR      60$          ;EXIT THE TEST.
5309
5310          ;REPORT THAT RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO.
5311 026630 012767 001133 155126 8$:      MOV      #0603.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5312 026636 012701 010033          MOV      #EM0603,R1      ;SELECT ERROR MESSAGE.
5313 026642          ERROR          ;REPORT ERROR.          >>>>> ERROR #0603 <<<<<
          026642 104460          TRAP      C#ERROR
5314
5315 026644          60$:      SETPRI      #PRI07          ;DISABLE ALL INTERRUPTS.
          026644 012700 000340          MOV      #PRI07,R0
          026650 104441          TRAP      C#SPRI
5316 026652 005067 153424          CLR      CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5317 026656          ENDTST
          026656 104401          L10033:
          TRAP      C#ETST

```

```
5319 .SBTTL HARDWARE TEST - RDVA -
5320 ;+ *****
5321 ;* - RBUF RX DATA VALID BIT TEST -
5322 ;* THIS TEST VERIFIES THAT THE DUT RBUF RX DATA VALID BIT IS SET BY THE
5323 ;* INCLUSION OF THE SELFTEST CODES IN THE DUT FIFO AND THAT THE BIT CLEARS
5324 ;* AFTER THE FIFO HAS BEEN EMPTIED.
5325 ;*
5326 ;-- *****
5327 026660 BGNTST
5328 026660 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T7::
5329 026660 012700 000240 MOV #PRI05,R0
5330 026664 104441 TRAP C$SPRI
5331 000007 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5332 026666 012767 000007 153424 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (7)
5333 026674 012767 177777 153400 MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5334 026702 012767 000001 155052 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5335 026710 012767 010216 155050 MOV #EM0701,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5336 026716 012767 015776 155044 MOV #ER0201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5337 ;+
5338 ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5339 ; AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5340 ;--
5341 026724 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5342 026730 012702 000040 MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5343 026734 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5344 026736 016704 153274 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5345 026742 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5346 026744 004767 173330 JSR PC,SKPSTS ;SKIP THE SELFTEST.
5347 026750 004767 171144 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5348 026754 103012 BCC 4$ ;GO REPORT ERROR IF MR DID NOT CLEAR.
5349 ;+
5350 ; CHECK THAT THE RX DATA VALID BIT IS SET.
5351 ;--
5352 026756 012400 MOV (R4)+,R0 ;INCREMENT POINTER TO PNT TO DUT RBUF REG.
5353 026760 005714 TST (R4) ;TEST THE DUT RX.DATA.VALID BIT.
5354 026762 100016 BPL 6$ ;GO REPORT ERROR IF BIT IS NOT SET.
5355 ;+
5356 ; READ CHARACTERS FROM THE DUT RX FIFO AND WAIT FOR RX.DATA.VALID TO GO CLEAR.
5357 ;--
5358 026764 012705 001130 MOV #600.,R5 ;ALLOW READING 600 CHARS BEFORE ERROR.
5359 026770 011400 2$: MOV (R4),R0 ;READ A CHARACTER FROM THE RX FIFO.
5360 026772 100027 BPL 60$ ;EXIT TEST WITHOUT ERROR IF BIT IS CLEAR.
5361 026774 005305 DEC R5 ;COUNT THE CHARACTER JUST READ.
5362 026776 001374 BNE 2$ ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
5363 027000 000416 BR 8$ ;GO REPORT ERROR IF RX.DATA.VALID WOULDN'T CLR.
5364 ;+
5365 ; ERROR REPORTS:
5366 ;--
5367 027002 012767 001275 154754 4$: ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5368 027010 012701 005665 MOV #0701.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5369 027014 104460 MOV #EM0202,R1 ;SELECT ERROR MESSAGE.
5370 027016 000415 ERROR ;REPORT ERROR. >>>> ERROR #0701 <<<<
5371 TRAP C$ERROR ;EXIT THE TEST.
```

```

5372
5373 027020 012767 001276 154736 61: ;REPORT THAT RX.DATA.VALID BIT WAS NOT SET AFTER A RESET COMPLETION.
5374 027026 012701 010261          MOV    #0702.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5375 027032          MOV    #EM0702,R1 ;SELECT ERROR MESSAGE.
          ERROR          ;REPORT ERROR.          >>>>> ERROR #0702 <<<<<
          027032 104460          TRAP    C#ERROR
5376 027034 000406          BR      601          ;EXIT THE TEST.
5377
5378          ;REPORT THAT RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO.
5379 027036 012767 001277 154720 81:          MOV    #0703.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5380 027044 012701 010441          MOV    #EM0703,R1 ;SELECT ERROR MESSAGE.
5381 027050          ERROR          ;REPORT ERROR.          >>>>> ERROR #0703 <<<<<
          027050 104460          TRAP    C#ERROR
5382
5383 027052          601:      SETPFI #PRI07          ;DISABLE ALL INTERRUPTS.
          027052 012700 000340          MOV    #PRI07,R0
          027056 104441          TRAP    C#SPRI
5384 027060 005067 153216          CLR    CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5385 027064          ENDTST
          027064 104401          L10034:
          TRAP    C#ETST

```

```

5387 .SBTTL  HARDWARE TEST          - RLNA -
5388 ;* *****
5389 ;*          - RBUF RX LINE NUMBER FIELD TEST -
5390 ;*          THIS TEST VERIFIES THAT THE DUT RBUF RX LINE NUMBER FIELD IS WORKING
5391 ;*          CORRECTLY BY UTILIZING THE SELFTEST CODES WHICH ARE PUT IN THE RX
5392 ;*          FIFO AFTER A BOARD RESET.
5393 ;* *****
5394 ;*
5395 027066 BGNTST
5396 027066          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.          T8::
5397 027066          MOV          #PRI05,R0          TRAP          C#SPRI
5398 027072 104441          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5399 027074 012700 000240          MOV          #TNUM,TSTNUM          ;SET UP THE TEST NUMBER.          (8)
5400 027102 012767 177777 153216          MOV          #-1,CTRLCF          ;INDICATE THAT WE ARE WITHIN A TEST.
5401 027110 012767 000001 154644          MOV          #1,ERRTYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5402 027116 012767 010624 154642          MOV          #EM0801,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5403 ;*
5404 ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5405 ; AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5406 ;*
5407 027124 012701 011610          MOV          #5000.,R1          ;TIME-OUT VALUE IS 5.0 SECONDS.
5408 027130 012702 000040          MOV          #BIT05,R2          ;WAITING FOR MASTER RESET BIT.
5409 027134 005003          CLR          R3          ;WAITING FOR BIT TO CLEAR.
5410 027136 016704 153074          MOV          CSRA,R4          ;BIT IS IN THE DUT'S CSR.
5411 027142 010214          MOV          R2,(R4)          ;SET THE DUT MASTER RESET BIT.
5412 027144 004767 173130          JSR          PC,SKPSTS          ;SKIP THE SELFTEST.
5413 027150 004767 170744          JSR          PC,MSLGET          ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5414 027154 103016          BCC          #          ;GO REPORT ERROR IF MR DID NOT CLEAR.
5415 ;*
5416 ; READ CHARACTERS FROM THE DUT RX FIFO AND VERIFY THAT THE LINE NUMBERS ARE
5417 ; CORRECT.
5418 ; EIGHT CHARACTERS ARE READ FROM THE FIFO.
5419 ;*
5420 027156 005001          CLR          R1          ;CLEAR THE LINE COUNTER.
5421 027160 012400          MOV          (R4),R0          ;INCREMENT POINTER TO PNT TO THE DUT RBUF REG.
5422 027162 011402          MOV          (R4),R2          ;READ A CHARACTER FROM THE DUT RX FIFO.
5423 027164 010203          MOV          R2,R3
5424 027166 000303          SWAB          R3
5425 027170 042703 177760          BIC          #177760,R3          ;REMOVE ALL BUT LINE NUMBER BITS.
5426 027174 020301          CMP          R3,R1          ;COMPARE WITH EXPECTED LINE NUMBER.
5427 027176 001017          BNE          #          ;GO REPORT ERROR IF LINE NUMBERS DON'T MATCH.
5428 027200 005201          INC          R1          ;INCREMENT THE EXPECTED LINE NUMBER.
5429 027202 020127 000010          CMP          R1,#8          ;COMPARE NUMBER OF CODES READ WITH MAX.
5430 027206 001365          BNE          #          ;LOOP UNTIL CODES FOR ALL LINES ARE READ.
5431 027210 000423          BR          #60          ;EXIT TEST WITHOUT ERROR.
5432 ;*
5433 ; ERROR REPORTS:
5434 ;*
5435 ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5436 027212 012767 001441 154544          4: MOV          #0801.,ERRNBR          ;SET THE ERROR NUMBER IN ERROR TABLE.
5437 027220 012767 016072 154542          MOV          #ER0503,ERRBLK          ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5438 027226 012701 005665          MOV          #EM0202,R1          ;SELECT ERROR MESSAGE.
5439 027232          ERROR          ;REPORT ERROR.          >>>>> ERROR #0801 <<<<<
5439 027232 104460          TRAP          C#ERROR

```

```

5440 027234 000411          BR      604          ;EXIT THE TEST.
5441
5442
5443 027236 012767 001442 154520 61: ;REPORT THAT RX LINE NUMBER FIELD IS WRONG FOR SELFTEST CODE.
5444 027244 012767 016072 154516      MOV    #0802,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5445 027252 012701 010672      MOV    #ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5446 027256      MOV    #EM0802,R1 ;SELECT ERROR MESSAGE.
5447 027256 104460      ERROR      ;REPORT ERROR. >>>>> ERROR #0802 <<<<<
5448      TRAP      C#ERROR
5449 027260      604:      SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
5450 027260 012700 000340      MOV    #PRI07,RO
5451 027264 104441      TRAP      C#SPRI
5452 027266 005067 153010      CLR     CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5453 027272      ENDTST
5454 027272 104401      L10035:
5455      TRAP      C#ETST

```

```

5452 .SBTTL  HARDWARE TEST          - BMPCHK -
5453 ;-- *****
5454 ;*          - BMP CHECK TEST -
5455 ;*      THIS TEST IS USED TO VERIFY THAT THE DUT DOES NOT IMMEDIATELY FAIL
5456 ;*      THE ON-BOARD BACKGROUND-MONITOR PROGRAM, AND HENCE INVALIDATE
5457 ;*      SUCCEEDING TESTS.
5458 ;*      THIS TEST LOOKS FOR BMP CODES IN THE FIFO FOR A SET PERIOD IMMEDIATELY
5459 ;*      AFTER THE SELF-TEST IS SKIPPED.
5460 ;*      ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE QUEUE AND ARE ALSO
5461 ;*      REPORTED IN THIS TEST.
5462 ;*
5463 ;-- *****
5464 027274      BGNTST
5465 027274      SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.          T9::
5466 027274      012700 000240          ;MOV      #PRI05,R0
5467 027300      104441          ;TRAP    C$SPRI
5468 027302      012767 000011 153010 ;TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5469 027310      012767 177777 152764 ;MOV      #TNUM,TSTNUM      ;SET UP THE TEST NUMBER.          (9)
5470 027316      012767 000001 154436 ;MOV      #-1,CTRLCF        ;INDICATE THAT WE ARE WITHIN A TEST.
5471 027324      012767 001605 154432 ;MOV      #1,ERRTYP         ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5472 027332      012767 010742 154426 ;MOV      #0901,ERRPNBR     ;SET THE ERROR NUMBER.
5473 ;*
5474 ;*      WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5475 ;*      IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5476 027340      012701 005670          ;MOV      #3000.,R1         ;TIME-OUT VALUE IS 3.0 SECONDS.
5477 027344      012702 000040          ;MOV      #BIT05,R2        ;WAITING FOR MASTER RESET BIT.
5478 027350      005003          ;CLR      R3               ;WAITING FOR BIT TO CLEAR.
5479 027352      016704 152660          ;MOV      CSRA,R4          ;BIT IS IN THE DUT'S CSR.
5480 027356      004767 170536          ;JSR      PC,MSLGET        ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5481 027362      103027          ;BCC      504                ;ABORT THE TEST IF MR DID NOT CLEAR.
5482 ;*
5483 ;*      RESET THE DUT, SKIP THE S'LF-TEST.
5484 ;*
5485 027364      010214          ;MOV      R2,(R4)          ;SET THE DUT MASTER RESET BIT.
5486 027366      004767 172706          ;JSR      PC,SKPST         ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
5487 ;*
5488 ;*      WAIT FOR MASTER RESET 0 CLEAR. DELAY FOR 500 MILLI-SECS BEFORE PURGING
5489 ;*      THE FIFO.
5490 ;*
5491 027372      012704 000764          ;MOV      #500.,R4         ;TIME-OUT VALUE IS 500 MILLI-SECONDS.
5492 027376      004767 170456          ;JSR      PC,DELAY         ;WAIT FOR BMP TO BEGIN EXECUTION.
5493 027402      004767 171070          ;JSR      PC,PUFIFO        ;PURGE THE FIFO, SAVING ANY BMP CODES.
5494 027406      103015          ;BCC      504                ;ABORT THE TEST IF THE FIFO DID NOT CLEAR.
5495 ;*
5496 ;*      REPORT THE ERROR IF ANY BMP CODES WERE FOUND.
5497 ;*
5498 027410      016702 153104          ;MOV      BMPCQP,R2        ;GET THE CONTENTS OF THE POINTER TO THE BMP Q.
5499 027414      012703 002522          ;MOV      #BMPCQB,R3       ;GET THE START ADDRESS OF THE QUEUE.
5500 027420      020203          ;CMP      R2,R3            ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
5501 027422      001414          ;BEQ      604                ;EXIT NO CODES IN THE QUEUE.
5502 ;*
5503 ;*      THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
5504 ;*
5505 ;*      ,REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN"

```

Address	Instruction	Op-Code	Register	Immediate	Comments
5506	MOV	027424	#EM0902,R1	012701 011002	PASS THE MESSAGE TO BE REPORTED.
5507	ERRDF	027430	0901,EM0901,ER9301		ERROR #0901 <<<<<.
		027430	104455		TRAP C#ERDF
		027432	001605		.WORD 901
		027434	010742		.WORD EM0901
		027436	017216		.WORD ER9301
5508	BR	027440	604	000405	
5509					
5510	MOV	027442	#902,ERRNBR	012767 001606 154314 504:	SET >>>> ERROR #0902 <<<<<.
5511	JSR	027450	PC,TSABRT	004767 172752	REPORT NON-TEST RELATED ERROR.
5512					
5513	SETPRI	027454	#PRI07	604:	DISABLE ALL INTERRUPTS.
		027454		012700 000340	
		027460		104441	MOV #PRI07,R0
5514	CLW	027462	CTRLCF	005067 152614	INDICATE THAT WE COMPLETED THE TEST.
5515	ENDTST	027466			
		027466			L10036:
		027466		104401	TRAP C#ETST

```

5517
5518
5519 .SBTTL  HARDWARE TEST          - SKSELF -
5520 ;* *****
5521 ;*          - SKIP SELF-TEST TEST -
5522 ;*          THIS TEST VERIFIES THAT THE DUT SKIPS THE SELF-TEST WITHIN THE
5523 ;*          TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS
5524 ;*          COMPLETION.
5525 ;*
5526 ;* *****
5527 027470 BGNTST
5528 027470          SETPRI  #PRI05          ,ALLOW LTC INTERRUPTS.          T10::
5529 027470          012700  000240          MOV          #PRI05,R0
5530 027474          104441          TRAP          C#SPRI
5531 027476          012767  000012  152614          TNUM -- TNUM + 1          ,INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5532 027504          012767  177777  152570          MOV          #TNUM,TSTNUM          ,SET UP THE TEST NUMBER. (10)
5533 027512          012767  000001  154242          MOV          #-1,CTRLCF          ,INDICATE THAT WE ARE WITHIN A TEST.
5534 027520          012767  011036  154240          MOV          #1,ERRTYP          ,SET ERROR TYPE AS FATAL IN ERROR TABLE.
5535 027526          012767  016072  154234          MOV          #EM1001,ERRMSG          ,SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5536          MOV          #ER0503,ERRBLK          ,SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5537 ;*
5538 ;* WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5539 ;* IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5540 ;*
5541          MOV          #5000.,R1          ,TIME-OUT VALUE IS 5.0 SECONDS.
5542          MOV          #BIT05,R2          ,WAITING FOR MASTER RESET BIT.
5543          CLR          R3          ,WAITING FOR BIT TO CLEAR.
5544          MOV          CSRA,R4          ,BIT IS IN THE DUT'S CSR.
5545          JSR          PC,MSLGET          ,WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5546          BCC          500          ,ABORT THE TEST IF MR DID NOT CLEAR.
5547 ;*
5548 ;* DETERMINE IF THE DUT TAKES TOO SHORT OR TOO LONG A TIME TO SKIP THE SELF-TEST
5549 ;* SET-UP A TIME-OUT OF 50 MILLI-SECOND, IF MR IS CLEAR IN LESS THAN 10 MILLI
5550 ;* -SECOND, OR GREATER THAN 50 MILLI-SECONDS, REPORT THE ERROR.
5551 ;*
5552          MOV          #50.,R1          ,TIME-OUT VALUE IS 50 MILLI-SECONDS.
5553          MOV          R2,(R4)          ,SET THE DUT MASTER RESET BIT.
5554          JSR          PC,SKPSTS          ,WRITE THE SKIP SELFTEST CODES TO THE DUT.
5555          JSR          PC,MSLGET          ,WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5556          BCC          200          ,GO REPORT ERR IF SKIPPING STEST TOOK TOO LONG.
5557          CMP          R1,#40.          ,GO REP ERR IF SELFTEST COMPLETED IN < 10 MS.
5558          BGT          400
5559 ;*
5560 ;* SELF-TEST COMPLETED WITHIN 10 MILLI-SEC TO 50 MILLI-SECONDS.
5561 ;* VERIFY THAT THE SELF-TEST CODES IN THE FIFO ARE "GOOD" CODES ,IE THE DUT
5562 ;* SUCCESSFULLY COMPLETED THE SELF-TEST.
5563 ;* THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 1003 THRU 1007 <<<<<.
5564 ;*
5565          MOV          #1003.,ERRNBR          ,SET ERROR NUMBER TO 1003.
5566          JSR          PC,RSTRPT          ,CHECK SELF-TEST CODES IN THE FIFO.
5567          BR          600          ,EXIT TEST.
5568 ;*
5569 ;* ERROR REPORTS:
5570 ;*
5571          ,REPORT SKIP SELF-TEST TOOK TOO LONG.
5572          MOV          #1001.,ERRNBR          ,SET THE ERROR NUMBER IN THE ERROR TABLE.

```

```

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

```

```

5588
5589
5590 .SBTTL  HARDWARE TEST          - DFSKST -
5591 ;* *****
5592 ;*          - DIAGNOSTIC FAIL BIT, SKIP SELF-TEST TEST -
5593 ;*          THIS TEST VERIFIES THAT THE DIAGNOSTIC FAIL BIT OF THE DUT, CORRECTLY
5594 ;*          CHANGES STATE AS THE ON-BOARDED SELFTEST IS SKIPPED.
5595 ;*
5596 ;* *****
5597 027704 BGNTST
5598 027704          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.          T11::
5599 027704          012700 000240          ;
5600 027710          104441          ;
5601 027710          000013          ;
5602 027712          012767 000013 152400          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5603 027720          012767 177777 152354          MOV  #TNUM,TSTNUM          ;SET UP THE TEST NUMBER.          (11)
5604 027726          012767 000001 154026          MOV  #-1,CTRLCF          ;INDICATE THAT WE ARE WITHIN A TEST.
5605 027734          012767 011234 154024          MOV  #1,ERRTYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5606 027742          012767 016072 154020          MOV  #EM1101,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5607          ;
5608          ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5609          ; IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5610          ;
5611          ;
5612          MOV  #5000.,R1          ;TIME-OUT VALUE IS 5.0 SECONDS.
5613          MOV  #BIT05,R2          ;WAITING FOR MASTER RESET BIT.
5614          CLR  R3          ;WAITING FOR BIT TO CLEAR.
5615          MOV  CSRA,R4          ;BIT IS IN THE DUT'S CSR.
5616          JSR  PC,MSLGET          ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5617          BCC  50$          ;ABORT THE TEST IF MR DID NOT CLEAR.
5618          ;
5619          ; RESET THE DUT, SKIP THE SELF-TEST.
5620          ;
5621          MOV  R2,(R4)          ;SET THE DUT MASTER RESET BIT.
5622          JSR  PC,SKPSTS          ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
5623          ;
5624          ; SET TIME OUT OF 5 MILLI SECONDS, WAIT FOR DIAG_FAIL BIT TO SET.
5625          ; IF TIME-OUT OCCURS GO REPORT THE ERROR.
5626          ;
5627          ;
5628          MOV  #5,R1          ;TIME-OUT VALUE IS 5 MILLI-SECONDS.
5629          MOV  #BIT13,R2          ;WAITING FOR DIAGNOSTIC FAIL BIT.
5630          MOV  R2,R3          ;WAITING FOR BIT TO SET.
5631          MOV  CSRA,R4          ;BIT IS IN THE DUT'S CSR.
5632          JSR  PC,MSLGET          ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
5633          BCC  4$          ;IF DIAG_FAIL DID NOT SET, GO REPORT ERROR.
5634          ;
5635          ; SET TIME-OUT OF 15 MILLI-SECS, WAIT FOR DIAG_FAIL TO CLEAR.
5636          ; IF TIME-OUT OCCURS GO REPORT THE ERROR.
5637          ; VERIFY THE DIAG FAIL BIT IS IN A STABLE STATE BEFORE CONTINUING. LOOP
5638          ; BACK IF THE STATE WAS TRANSITORY, USING THE REMAINDER OF THE 15 MS TIME-OUT.
5639          ;
5640          ;
5641          MOV  #15.,R1          ;TIME-OUT VALUE IS 15 MILLI-SECONDS.
5642          CLR  R3          ;WAITING FOR BIT TO CLEAR.
5643          JSR  PC,MSLGET          ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
5644          BCC  4$          ;IF DIAG_FAIL DID NOT CLEAR, GO REPORT ERROR.
5645          MOV  R1,R5          ;SAVE THE REMAINING TIME-OUT VALUE.

```

```

5642 030044 012701 000001      MOV    #1,R1      ;SET TIME-OUT OF 1 MILLI-SECOND.
5643 030050 052703 020000      BIS    #BIT13,R3    ;WAIT FOR BIT TO SET.
5644 030054 004767 170040      JSR    PC,MSLGET    ;DOUBLE CHECK TO ELIMINATE NOISE PROBLEMS.
5645 030060 103016              BCC    60$          ;EXIT IF DIAG_FAIL BIT STILL CLEAR.
5646 030062 010501              MOV    R5,R1      ;PASS THE REMAINING TIME-OUT VALUE.
5647 030064 000762              BR     2$          ;LOOP TO CHECK AGAIN.
5648
5649                      ;
5650                      ; ERROR REPORTS:
5651                      ;
5652 030066 012767 002115 153670 4$:      ;REPORT DIAGNOSTIC FAIL BIT BAD.
5653 030074 012701 011533              MOV    #1101.,ERRNBR    ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5654 030100 104460              MOV    #EM1205,R1    ;SELECT ERROR MESSAGE.
5655 030102 000405              ERROR              ;REPORT ERROR.          >>>>> ERROR #1101 <<<<<
5656                      BR     60$              TRAP    C$ERROR
5657 030104 012767 002116 153652 50$:      ;EXIT THE TEST.
5658 030112 004767 172310              MOV    #1102.,ERRNBR    ;SET THE ERROR NUMBER FOR TSABRT RTN.
5659                      JSR    PC,TSABRT          ;REPORT NON-TEST RELATED ERROR.
5660 030116 012700 000340 60$:      SETPRI  #PRI07      ;DISABLE ALL INTERRUPTS.
5661 030124 005067 152152              MOV    #PRI07,R0    ;
5662 030130 104401              TRAP    C$SPRI
5663 030130 000000              CLR    CTRLCF      ;INDICATE THAT WE COMPLETED A TEST.
5664 030130 000000              ENDTST
5665 030130 104401              L10040:
5666 030130 000000              TRAP    C$ETST

```

```

5664 .SBTTL HARDWARE TEST - SELFTS -
5665 ;*****
5666 ;* - SELF-TEST TEST -
5667 ;* THIS TEST VERIFIES THAT THE DUT'S SELF-TEST EXECUTES WITHIN THE
5668 ;* TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS
5669 ;* COMPLETION.
5670 ;*
5671 ;-- *****
5672 030132 BGNTST
5673 030132 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T12::
5674 030132 012700 000240 MOV #PRI05,RO
5675 030132 104441 TRAP C#SPRI
5676 030132 000014 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5677 030140 012767 000014 152152 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (12)
5678 030146 012767 177777 152126 MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5679 030154 012767 000001 153600 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5680 030162 012767 011307 153576 MOV #EM1201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5681 030170 012767 016072 153572 MOV #ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5682 ;*
5683 ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5684 ; IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5685 ;--
5686 030176 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5687 030202 012702 000040 MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5688 030206 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5689 030210 016704 152022 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5690 030214 004767 167700 JSR PC,MSLGET ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5691 030220 103067 BCC 504 ;ABORT THE TEST IF MR DID NOT CLEAR.
5692 ;*
5693 ; DETERMINE IF THE SELF-TEST TAKES TOO SHORT OR TOO LONG A TIME TO COMPLETE.
5694 ; SET-UP A TIME-OUT OF 5 SECOND, IF MR IS CLEAR IN LESS THAN 1/2 SECOND, OR
5695 ; GREATER THAN 5 SECONDS, REPORT THE ERROR.
5696 ;--
5697 030222 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5698 030226 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5699 030230 004767 167664 JSR PC,MSLGET ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5700 030234 103034 BCC 41 ;GO REPORT ERROR SELFTEST TOOK TOO LONG.
5701 030236 012702 011610 MOV #5000.,R2
5702 030242 160102 SUB R1,R2 ;CALCULATE # OF MS SELFTEST TO COMPLETE.
5703 030244 020227 000062 CMP R2,#50.
5704 030250 002435 BLT 61 ;SELFTEST SKIPPED? YES, GO REPORT ERROR.
5705 030252 020227 000764 CMP R2,#500.
5706 030256 002441 BLT 81 ;GO REP ERR IF SELFTEST COMPLETED IN < 1/2 SEC.
5707 ;*
5708 ; SELF-TEST COMPLETED WITHIN 1SEC TO 5 SECONDS.
5709 ; CHECK THE STATE OF THE DIAGNOSTIC FAIL BIT, REPORT ERROR IF IT IS SET.
5710 ;--
5711 030260 032714 020000 BIT #BIT13,(R4) ;DETERMINE IF THE DIAG FAIL BIT IS CLEAR.
5712 030264 001412 BEQ 21 ;SKIP ERROR REPORT IF BIT IS CLEAR.
5713 030266 012767 002264 153470 ;REPORT DIAGNOSTIC FAIL BIT BAD.
5714 030274 012701 011533 MOV #1204.,ERRNBR ;SET ERROR NUMBER TO IN ERROR TABLE.
5715 030300 012701 011533 MOV #EM1205,R1 ;SELECT THE ERROR MESSAGE.
5716 030300 104460 ERROR ;>>>> ERROR #1204 <<<<<
; TRAP C#ERROR
; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

```

```

5717
5718 030302 032767 000100 151710      BIT      #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
5719 030310 001440                      BEQ      60#      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5720                                      ;DURING THE SOFTWARE QUESTIONS.
5721
5722      ;*
5723      ; VERIFY THAT THE SELF-TEST CODES IN THE FIFO ARE "GOOD" CODES ,IE THE DUT
5724      ; SUCCESSFULLY COMPLETED THE SELF-TEST.
5725      ; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 1205 THRU 1209 <<<<<.
5726 030312 012767 002265 153444 2# :      MOV      #1205.,ERRNBR ;SET ERROR NUMBER TO 1205.
5727 030320 004767 171124          JSR      PC,RSTRPT ;CHECK SELF-TEST CODES IN THE FIFO.
5728 030324 000432          BR      60#      ;EXIT TEST.
5729
5730      ;*
5731      ; ERROR REPORTS:
5732      ;-
5733 030326 012767 002261 153430 4# :      ;REPORT SELF-TEST TOOK TOO LONG TO COMPLETE.
5734 030334 012701 011335          MOV      #1201.,ERRNBR ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5735 030340          MOV      #EM1202,R1 ;SELECT ERROR MESSAGE.
5736 030342 104460          ERROR          ;REPORT ERROR. >>>>> ERROR #1201 <<<<<
5737          BR      60#      TRAP      C#ERROR
5738          ;EXIT THE TEST.
5739 030344 012767 002262 153412 6# :      ;REPORT SELF-TEST DID NOT EXECUTE AFTER DUT RESET.
5740 030352 012701 011477          MOV      #1202.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5741 030356          MOV      #EM1204,R1 ;SELECT ERROR MESSAGE.
5742 030360 104460          ERROR          ;REPORT ERROR. >>>>> ERROR #1202 <<<<<
5743          BR      60#      TRAP      C#ERROR
5744          ;EXIT THE TEST.
5745 030362 012767 002263 153374 8# :      ;REPORT SELF-TEST COMPLETED TOO SOON.
5746 030370 012701 011421          MOV      #1203.,ERRNBR ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5747 030374          MOV      #EM1203,R1 ;SELECT ERROR MESSAGE.
5748 030376 104460          ERROR          ;REPORT ERROR. >>>>> ERROR #1203 <<<<<
5749          BR      60#      TRAP      C#ERROR
5750          ;EXIT THE TEST.
5750 030400 012767 002272 153356 50# :      MOV      #1210.,ERRNBR ;SET THE ERROR NUMBER FOR TSABRT RTN.
5751 030406 004767 172014          JSR      PC,TSABRT ;REPORT NON-TEST RELATED ERROR.
5752
5753 030412          60# :      SETPRI   #PRI07 ;DISABLE ALL INTERRUPTS.
5754 030416 012700 000340          MOV      #PRI07,R0
5755 030420 005067 151656          CLR      CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
5755 030424          ENDTST
5755 030424 104401          L10041: TRAP      C#ETST

```

```

5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767 030426
      030426
5768 030426
      030426 012700 000240
      030432 104441
5769      000015
5770 030434 012767 000015 151656
5771 030442 012767 177777 151632
5772 030450 012767 000001 153304
5773 030456 012767 011557 153302
5774 030464 012767 016072 153276
5775 030472 012767 002425 153264
5776
5777
5778
5779
5780 030500 012701 011610
5781 030504 012702 000040
5782 030510 005003
5783 030512 016704 151520
5784 030516 004767 167376
5785 030522 103064
5786
5787
5788
5789
5790
5791 030524 012777 000040 151504
5792 030532 012704 000031
5793 030536 004767 167316
5794 030542 012777 146314 151504
5795
5796
5797
5798
5799 030550 005267 153210
5800 030554 012701 011610
5801 030560 012702 000040
5802 030564 005003
5803 030566 016704 151444
5804 030572 004767 167322
5805 030576 103036
5806
5807
5808
5809
5810 030600 005267 153160

.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.
;+ *****
;-- *****
      BGNTST
                                T13::
      SETPRI  #PRI05              ;ALLOW LTC INTERRUPTS.
                                MOV      #PRI05,R0
                                TRAP     C#SPRI
      TNUM == TNUM + 1           ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV      #TNUM,TSTNUM      ;SET UP THE TEST NUMBER. (13)
      MOV      #-1,CTRLCF        ;INDICATE THAT WE ARE WITHIN A TEST.
      MOV      #1,ERRTYP         ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
      MOV      #EM1301,ERRMSG     ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
      MOV      #ER0503,ERRBLK    ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
      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.
;--
      MOV      #5000.,R1          ;TIME-OUT VALUE IS 5.0 SECONDS.
      MOV      #BIT05,R2          ;WAITING FOR MASTER RESET BIT.
      CLR      R3                 ;WAITING FOR BIT TO CLEAR.
      MOV      CSRA,R4            ;BIT IS IN THE DUT'S CSR.
      JSR      PC,MSLGET          ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
      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 TBUFFT REGISTER ON CHANNEL 0.
;--
      MOV      #BIT05,BCSRA       ;SET DUT MASTER RESET BIT, SELECT CHANNEL 0.
      MOV      #25.,R4            ;PASS DELAY PERIOD OF 25 MILLI SECS.
      JSR      PC,DELAY           ;WAIT FOR SELFTEST TO INITIALISE.
      MOV      #146314,BTXBFC     ;WRITE THE FAIL SELF-TEST CODE TO TBUFFT REG.
;+
;+  WAIT UP TO 5 SECONDS FOR THE SELF-TEST TO COMPLETE.
;+  IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
;--
      INC      ERRNBR             ;SET ERROR NUMBER TO 1302.
      MOV      #5000.,R1          ;TIME-OUT VALUE IS 5.0 SECONDS.
      MOV      #BIT05,R2          ;PASS THE BIT MAP OF THE BIT TO TEST.
      CLR      R3                 ;SET UP THE EXPECTED STATE.
      MOV      CSRA,R4            ;BIT IS IN THE DUT'S CSR.
      JSR      PC,MSLGET          ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
      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.
;--
      INC      ERRNBR             ;SET ERROR NUMBER TO 1303.

```

5811	030604	032714	020000		BIT #BIT13,(R4)	;CHECK THE STATE OF THE DIAG_FAIL BIT.	
5812	030610	001425			BEQ 10\$	;GO REPORT ERROR IF DIAG_FAIL BIT CLEAR.	
5813							
5814							
5815							
5816							
5817	030612	005267	153146		INC ERRNBR	;SET ERROR NUMBER TO 1304.	
5818	030616	012700	000010		MOV #8.,R0	;SET MAXIMUM READ COUNT.	
5819	030622	005001			CLR R1	;CLEAR THE CORRECT CODE COUNTER.	
5820	030624	016704	151410		MOV RBUFA,R4	;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.	
5821	030630	011402		6\$:	MOV (R4),R2	;READ A CODE FROM THE FIFO.	
5822	030632	100020			BPL 50\$	;GO REPORT ERROR IF THE FIFO IS EMPTY.	
5823	030634	042702	007400		BIC #7400,R2	;REMOVE THE LINE NUMBER FROM THE CODE.	
5824	030640	120227	170231		CMPB R2,#170231	;IS IT THE CORRECT ERROR CODE?.	
5825	030644	001001			BNE 8\$	;SKIP NEXT INSTRUCTION, IF NOT A 231 CODE.	
5826	030646	005201			INC R1	;INCREMENT COUNTER.	
5827	030650	005300		8\$:	DEC R0	;DECREMENT MAX READ COUNTER.	
5828	030652	001366			BNE 6\$	;LOOP IF 8 CODES HAVE NOT BEEN READ.	
5829	030654	005701			TST R1	;WERE ANY 231 CODES FOUND?.	
5830	030656	001010			BNE 60\$	;YES, THEN EXIT.	
5831	030660	005267	153100		INC ERRNBR	;NO, SET ERROR NUMBER TO 1305 AND REPORT ERROR.	
5832						;REPORT SELF-TEST ERROR REPORTING BAD.	
5833	030664	012701	011612	10\$:	MOV #EM1302,R1	;SELECT ERROR MESSAGE.	
5834	030670				ERROR	;REPORT ERROR.	>>>> ERROR <<<< TRAP C#ERROR
5835	030672	104460			BR 60\$	;EXIT THE TEST.	
5836		000402					
5837	030674	004767	171526	50\$:	JSR PC,TSABRT	;REPORT NON-RELATED TEST ERROR.	
5838							
5839	030700			60\$:	SETPRI #PRI07	;DISABLE ALL INTERRUPTS.	
	030700	012700	000340				MOV #PRI07,R0 C#SPRI
	030704	104441					
5840	030706	005067	151370		CLR CTRLCF	;INDICATE THAT WE COMPLETED THE TEST.	
5841	030712				ENDTST		
	030712					L10042:	
	030712	104401				TRAP	C#ETST

```

5843
5844
5845 .SBTTL  HARDWARE TEST          - ROMVER -
5846 ;* *****
5847 ;*          - ROM VERSION TEST -
5848 ;* THIS TEST VERIFIES THAT THE DUT'S SELF-TEST PLACES VALID ROM VERSION
5849 ;* NUMBERS IN THE FIFO AFTER IT HAS BEEN SKIPPED.  THE ROM VERSION NUMBERS
5850 ;* WILL BE REPORTED (ON THE FIRST PASS ONLY), IF AN AFFIRMATIVE ANSWER
5851 ;* WAS GIVEN TO THE SOFTWARE P-TABLE QUESTION.
5852 ;* *****
5853 ;*
5854 030714      BGNTST
5855 030714      SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.      T14::
5856 030714      012700  000240          MOV      #PRI05,R0
5857 030720      104441          TRAP      C$SPRI
5858 030722      000016          TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5859 030722      012767  000016  151370      MOV      #TNUM,T$TNUM      ;SET UP THE TEST NUMBER.      (14)
5860 030730      012767  177777  151344      MOV      #-1,CTR$LCF      ;INDICATE THAT WE ARE WITHIN A TEST.
5861 030736      012767  000001  153016      MOV      #1,ERR$TYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5862 030744      012767  011651  153014      MOV      #EM1401,ERR$MSG      ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5863 030752      012767  016072  153010      MOV      #ER0503,ERR$BLK      ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5864 ;*
5865 ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5866 ; IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5867 ;*
5868 030760      012701  005670          MOV      #3000.,R1      ;TIME-OUT VALUE IS 3.0 SECONDS.
5869 030764      012702  000040          MOV      #BIT05,R2      ;WAITING FOR MASTER RESET BIT.
5870 030770      005003          CLR      R3      ;WAITING FOR BIT TO CLEAR.
5871 030772      016704  151240          MOV      CSRA,R4      ;BIT IS IN THE DUT'S CSR.
5872 030776      004767  167116          JSR      PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5873 031002      103131          BCC      $04      ;ABORT THE TEST IF MR DID NOT CLEAR.
5874 ;*
5875 ; SET THE MASTER RESET BIT, AND SKIP THE SELF TEST.
5876 ;*
5877 031004      010214          MOV      R2,(R4)      ;SET THE MASTER RESET BIT.
5878 031006      004767  171266          JSR      PC,SKP$STS      ;SKIP THE SELF TEST.
5879 031012      012701  011610          MOV      #5000.,R1      ;TIME-OUT VALUE IS 5.0 SECONDS.
5880 031016      004767  167076          JSR      PC,MSLGET      ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5881 031022      103121          BCC      $04      ;ABORT THE TEST IF MR DID NOT CLEAR.
5882 ;*
5883 ; REMOVE CHARACTERS FROM THE FIFO UNTIL EITHER,
5884 ; (A) THE FIFO IS PURGED, GO REPORT THE ERROR.
5885 ; (B) THE MAXIMUM TRY COUNTER IS ZERO, GO REPORT THE ERROR.
5886 ; (C) PROC_1'S ROM VERSION NUMBER WAS FOUND BEFORE PROC_2'S. GO REPORT ERROR.
5887 ; (D) BOTH ROM VERSION NUMBERS HAVE BEEN FOUND.
5888 ;*
5889 031024      012705  000040          MOV      #32.,R5      ;SET MAXIMUM TRY COUNTER.
5890 031030      012703  000143          MOV      #99.,R3      ;SET AN INVALID ROM VERSION NUMBER FOR PROC_1.
5891 031034      010304          MOV      R3,R4      ;SET AN INVALID ROM VERSION NUMBER FOR PROC_2.
5892 031036      012767  002571  152720      MOV      #1401.,ERR$NBR      ;SET THE ERROR NUMBER TO 1401.
5893 031044      012701  011710          MOV      #EM1402,R1      ;SELECT MESSAGE TO BE REPORTED IF FIFO EMPTY.
5894 031050      017702  151164      24:      MOV      $RBUFA,R2      ;READ THE NEXT CHAR FROM THE FIFO.
5895 031054      100077          BPL      124      ;GO REPORT ERROR IF FIFO EMPTY.
5896 ;*
5897 ; CHECK IF THE READ DATA IS A BMM CODE.

```

5897
5898 031056 012700 000301
5899 031062 040200
5900 031064 001003
5901 031066 004767 171140
5902 031072 000435
5903
5904
5905
5906 031074 012700 000201
5907 031100 040200
5908 031102 001431
5909
5910
5911
5912
5913 031104 032702 000002
5914 031110 001407
5915 031112 010204
5916 031114 042704 177603
5917 031120 000241
5918 031122 006004
5919 031124 006004
5920 031126 000417
5921 031130 010203
5922 031132 042703 177603
5923 031136 000241
5924 031140 006003
5925 031142 006003
5926 031144 020427 000143
5927 031150 001016
5928
5929
5930
5931
5932 031152 012701 011776
5933 031156 012767 002572 152600
5934 031164 000433
5935
5936 031166 005305
5937 031170 001327
5938 031172 012701 012051
5939 031176 012767 002573 152560
5940 031204 000423
5941
5942
5943
5944
5945 031206 032767 000001 151004
5946 031214 001431
5947 031216 026727 151064 000001
5948 031224 003025
5949 031226
031226 010446
031230 010346
031232 012746 004201
031236 012746 000003

MOV #301,R0 ;SET-UP A BIT MASK OF A BMP CODE.
BIC R2,R0 ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
BNE 41 ;BRANCH IF NOT A BMP CODE.
JSR PC,SAVBMP ;SAVE THE BMP CODE ON THE QUEUE.
BR 81
; CHECK IF THE READ DATA IS A SELF-TEST CODE.
41: MOV #201,R0 ;SET-UP A BIT MASK OF A SELFTEST CODE.
BIC R2,R0 ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
BEQ 81 ;BRANCH IF IT IS A SELFTEST CODE.
; THE READ DATA IS A ROM VERSION NUMBER, DETERMINE WHICH ONE IT IS.
; CHECK THE PROCESSOR NUMBER BIT IN THE CODE.
BIT #BIT1,R2 ;BRANCH IF IT IS PROC_1 ROM VERSION NUMBER.
BEQ 61 ;SAVE PROC_2 ROM VERSION NUMBER.
MOV R2,R4 ;CLEAR ANY UNWANTED BITS.
BIC #177603,R4 ;CLEAR THE CARRY BIT.
CLC ;SHIFT THE CODES ALONG TO GET THE ROM
ROR R4 ;VERSION NUMBER IN THE LOW 5 BITS.
ROR R4
BR 81
61: MOV R2,R3 ;SAVE PROC_1 ROM VERSION NUMBER.
BIC #177603,R3 ;CLEAR ANY UNWANTED BITS.
CLC ;CLEAR THE CARRY BIT.
ROR R3 ;SHIFT THE CODE ALONG TO GET THE ROM
ROR R3 ;VERSION NUMBER IN THE LOW 5 BITS.
CMP R4,#99. ;CHECK IF WE HAVE RECEIVE PROC_2 ROM CODE.
BNE 101 ;GO REPORT BOTH ROM VERSION NUMBERS.
; RECEIVED ROM VERSION NUMBERS OUT OF SEQUENCE.
; IE. PROC_1'S ROM VERSION NUMBER FOUND IN THE FIFO BEFORE PROC_2'S.
MOV #EM1403,R1 ;SELECT THE ERROR MESSAGE TO BE REPORTED.
MOV #1402.,ERRNBR ;SET THE ERROR NUMBER.
BR 121 ;GO REPORT ERROR.
81: DEC R5 ;DECREMENT THE MAX TRY COUNTER.
BNE 21 ;LOOP TO GET THE NEXT CHAR FROM THE FIFO.
MOV #EM1404,R1 ;SELECT THE ERROR MESSAGE TO BE REPORTED.
MOV #1403.,ERRNBR ;SET THE ERROR NUMBER.
BR 121 ;GIVE UP, GO REPORT ERROR.
; IF THIS IS THE FIRST PASS, AND SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
; THEN REPORT THE ROM VERSION NUMBERS TO THE OPERATOR.
101: BIT #BIT0,OPTION ;CHECK ON THE STATE OF THE SOFTWARE SWITCH.
BEQ 601 ;EXIT IF NO ROM VERSION PRINTOUT WAS REQUESTED.
CMP PASCNT,#1 ;CHECK IF THIS IS THE FIRST PASS.
BGT 601 ;EXIT IF ROM VERS HAVE ALREADY BEEN REPORTED.
PRINTB #EF1401,R3,R4 ;PRINT THE ROM VERSION NUMBERS.
MOV R4,-(SP)
MOV R3,-(SP)
MOV #EF1401,-(SP)
MOV #3,-(SP)

```

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

```

```

5965 .SBTTL  HARDWARE TEST          - REGWRW -
5966 ;* *****
5967 ;* - DEVICE REGISTER WORD ACCESS READ AND WRITE TEST -
5968 ;*
5969 ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
5970 ;* CORRECTLY USING WORD ACCESSES.
5971 ;*
5972 ;* -- *****
5973
5974 031314          BGNTST
5975 031314          SETPRI #PRI05          T15::
031314 012700 000240          ;ALLOW THE LTC TO INTERRUPT.
031320 104441          MOV #PRI05,R0
000017          TRAP C#SPRI
5976          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5977 031322 012767 000017 150770      MOV #TNUM,TSTNUM          ;SET UP THE TEST NUMBER. (16)
5978 031330 012767 177777 150744      MOV #-1,CTRLCF          ;INDICATE THAT WE ARE WITHIN A TEST.
5979 031336 012767 000001 152416      MOV #1,ERRTYP          ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
5980 031344 012767 003101 152412      MOV #1601,ERRNBR          ;SET UP ERROR NUMBER IN THE ERROR TABLE.
5981 031352 012767 012254 152406      MOV #EM1604,ERRMSG          ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
5982 031360 005067 151072          CLR ERSNRF          ;CLEAR THE ERROR SUMMARY FLAGS.
5983 031364 012700 002460          MOV #ERCNTB,R0
5984 031370 004767 166364          JSR PC,CLR16W          ;CLEAR THE ERROR COUNTER TABLE.
5985 031374 005067 150700          CLR EXOERR          ;CLEAR THE "EXIT ON ERROR" FLAG
5986
5987 ;*
5988 ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
5989 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
5990 ; THIS SUBROUTINE REPORTS ERRORS >>>> 1601 <<<<.
5991 031400 004767 167706          JSR PC,RESETT          ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
5992 031404 103402          BCS .+6          ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
5993 031406 000167 000142          JMP 601          ;YES, EXIT THE TEST.
5994
5995 ;*
5996 ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR
5997 031412 005267 152346          INC ERRNBR          ;SET THE ERROR REPORT NUMBER TO 1602.
5998 031416 012702 000017          MOV #17,R2          ;SET LOOP COUNT.
5999 031422 016704 150610          MOV CSRA,R4          ;GET CSR ADDRESS.
6000 031426 010214          MOV R2,(R4)          ;WRITE COUNT TO CSR.
6001 031430 011401          MOV (R4),R1          ;READ BACK THE CONTENTS OF THE CSR
6002 031432 042701 177760          BIC #177760,R1          ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
6003 031436 020102          CMP R1,R2          ;CHECK FOR CORRECT DATA WRITTEN/READ.
6004 031440 001412          BEQ 41          ;IS EXPECTED DATA BAD? NO, SKIP ERROR REPORT.
6005          ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE 0 (D)."
6006 031442 012767 016352 152320      MOV #ER1601,ERRBLK          ;SELECT THE PROPER ERROR REPORT ROUTINE.
6007 031450 005003          CLR R3          ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
6008 031452 005005          CLR R5          ;CAUSE REPORT OF LINE 0.
6009 031454          ERROR          ; >>>> ERROR # 1602 <<<<
031454 104460          TRAP C#ERROR
6010
6011 ;*
6012 ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
6013 031456 032767 000100 150534      BIT #BIT06,OPTION          ;HAS EXTENDED ERROR BEEN REQUESTED ?
6014 031464 001433          BEQ 601          ;EXIT THE TEST IF IT HASN'T.
6015
6016 41: DEC R2          ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
6017 031470 002356          BGE 21          ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.

```

```

6018
6019
6020
6021
6022
6023 031472 005267 152266
6024 031476 005003
6025 031500 012704 000002
6026 031504 004767 167322
6027
6028
6029
6030
6031 031510 005767 150564
6032 031514 001017
6033
6034
6035
6036
6037
6038 031516 012767 003106 152240
6039 031524 005003
6040 031526 005404
6041 031530 004767 167276
6042
6043
6044
6045
6046 031534 005767 150540
6047 031540 001005
6048
6049
6050
6051
6052 031542 012767 003111 152214
6053 031550 004767 167510
6054 031554 005067 150522
6055 031560
    031560
    031560 104401

;
; 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 <<<<.
;-
    INC     ERRNBR      ;SET THE ERROR NUMBER TO 1603.
    CLR     R3          ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
    MOV     #2,R4       ;INDICATE R/W ACCESS, CLEAR FIRST.
    JSR     PC,REGTST   ;WRITE AND VERIFY DATA PATTERNS.

;
; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
;-
    TST     EXOERR      ;IS THE "EXIT ON ERROR" FLAG SET ?
    BNE     601         ;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 <<<<.
;-
    MOV     #1606.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
    CLR     R3          ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
    NEG     R4          ;INDICATE R/W ACCESS, SET FIRST.
    JSR     PC,REGTST   ;WRITE AND VERIFY DATA PATTERNS.

;
; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
;-
    TST     EXOERR      ;IS THE "EXIT ON ERROR" FLAG SET ?
    BNE     601         ;EXIT IF IT IS.

;
; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
; THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>> ERROR # 1609 <<<<
;-
    MOV     #1609.,ERRNBR ;SET UP ERROR NUMBER FOR NEXT RTN.
    JSR     PC,REPSMR   ;REPORT ERROR SUMMARY IF NECESSARY.
    CLR     CTRLCF      ;INDICATE THAT WE COMPLETED THE TEST.
    ENDTST

601:

```

L10044: TRAP C#ETST

```
6057 .SBTTL HARDWARE TEST - REGWRM -
6058 ;* *****
6059 ;* - DEVICE REGISTER WORD ACCESS READ/MODIFY/WRITE TEST -
6060 ;*
6061 ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE WRITTEN CORRECTLY
6062 ;* USING WORD READ/MODIFY/WRITE ACCESSES.
6063 ;*
6064 ;* *****
6065 ;-- *****
6066 031562 BGNTST
6067 031562 T16::
031562 SETPRI #PRI05 ;ALLOW THE LTC TO INTERRUPT.
031562 012700 000240 MOV #PRI05,R0
031566 104441 TRAP C$SPRI
6068 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6069 031570 012767 000020 150522 MOV #TNUM,T$TNUM ;SET UP THE TEST NUMBER. (17)
6070 031576 012767 177777 150476 MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
6071 031604 012767 000001 152150 MOV #1,ERRTYP ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6072 031612 012767 003245 152144 MOV #1701,ERRNBR ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6073 031620 012767 012330 152140 MOV #EM1701,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6074 031626 005067 150624 CLR ERSRWF ;CLEAR THE ERROR SUMMARY FLAGS.
6075 031632 012700 002460 MOV #ERCNTB,R0
6076 031636 004767 166116 JSR PC,CLR16W ;CLEAR THE ERROR COUNTER TABLE.
6077 031642 005067 150432 CLR EXOERR ;CLEAR THE "EXIT ON ERROR" FLAG
6078 ;*
6079 ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6080 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6081 ; THIS SUBROUTINE REPORTS ERRORS >>>> 1701 <<<<.
6082 ;--
6083 031646 004767 167440 JSR PC,RESETT ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6084 031652 103402 BCS .+6 ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
6085 031654 000167 000064 JMP 601 ;YES, EXIT THE TEST.
6086 ;*
6087 ; THE READ/MODIFY/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR IS
6088 ; NOT TESTED THIS THIS FORM OF ACCESS IS ILLEGAL.
6089 ;--
6090 ;*
6091 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
6092 ; ACTIVE LINES USING R/M/W. BEFORE WRITING EACH PATTERN, CLEAR ALL THE BITS.
6093 ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1703 - 1705 <<<<.
6094 ;--
6095 031660 012767 003247 152076 MOV #1703,ERRNBR ;SET THE ERROR NUMBER TO 1703.
6096 031666 005003 CLR R3 ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6097 031670 012704 000001 MOV #1,R4 ;INDICATE R/M/W ACCESS, CLEAR FIRST.
6098 031674 004767 167132 JSR PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6099 ;*
6100 ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6101 ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6102 ;--
6103 031700 005767 150374 TST EXOERR ;IS THE "EXIT ON ERROR" FLAG SET ?
6104 031704 001017 BNE 601 ;EXIT IF IT IS.
6105 ;*
6106 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
6107 ; ACTIVE LINES USING R/M/W. BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
6108 ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1706 - 1708 <<<<.
6109 ;--
6110
```

```

6111
6112 031706 012767 003252 152050 ; MOV #1706.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6113 031714 005003 ; CLR R3 ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6114 031716 005404 ; NEG R4 ;INDICATE R/M/W ACCESS, SET FIRST.
6115 031720 004767 167106 ; JSR PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6116
6117 ;*
6118 ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6119 ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6120 031724 005767 150350 ;*
6121 031730 001005 ; TST EXOERR ;IS THE "EXIT ON ERROR" FLAG SET ?
6122 ; BNE 60$ ;EXIT IF IT IS.
6123
6124 ;*
6125 ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6126 ; THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>> ERROR # 1709 <<<<
6127 031732 012767 003255 152024 ;*
6128 031740 004767 167320 ; MOV #1709.,ERRNBR ;SET UP ERROR NUMBER FOR NEXT RTN.
6129 031744 005067 150332 ; JSR PC,REPSMR ;REPORT ERROR SUMMARY IF NECESSARY.
6.30 031750 60$: CLR CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
031750 ; ENDTST
031750 104401

```

L10045: TRAP C#ETST

```
6132 .SBTTL HARDWARE TEST - REGBRW -
6133 ;** *****
6134 ;* - DEVICE REGISTER BYTE ACCESS READ AND WRITE TEST -
6135 ;*
6136 ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
6137 ;* CORRECTLY USING BYTE ACCESSES.
6138 ;*
6139 ;-- *****
6140
6141 031752 BGNTST
6142 031752 SETPRI #PRI05 ;ALLOW THE LTC TO INTERRUPT.
6143 031752 012700 000240 ;MOV #PRI05,R0
6144 031752 104441 ;TRAP C#SPRI
6145 031752 000021 ;TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6146 031760 012767 000021 150332 ;MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (18)
6147 031766 012767 177777 150306 ;MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
6148 031774 012767 000001 151760 ;MOV #1,ERRTYP ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6149 032002 012767 003411 151754 ;MOV #1801.,ERRNBR ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6150 032010 012767 012413 151750 ;MOV #EM1801,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6151 032016 005067 150434 ;CLR ERSRWF ;CLEAR THE ERROR SUMMARY FLAGS.
6152 032022 012700 002460 ;MOV #ERCNTB,R0
6153 032026 004767 165726 ;JSR PC,CLR16W ;CLEAR THE ERROR COUNTER TABLE.
6154 032032 005067 150242 ;CLR EXOERR ;CLEAR THE "EXIT ON ERROR" FLAG.
6155
6156 ;*
6157 ;* RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6158 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6159 ;* THIS SUBROUTINE REPORTS ERRORS >>>> 1801 <<<<.
6160
6161 032036 004767 167250 ;JSR PC,RESETT ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6162 032042 103402 ;BCS .+6 ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
6163 032044 000167 000212 ;JMP 60 ;YES, EXIT THE TEST.
6164 032050 012767 003412 151706 ;MOV #1802.,ERRNBR ;SET THE ERROR REPORT NUMBER TO 1802.
6165
6166 ;*
6167 ;* VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR.
6168 ;* USE BYTE ACCESSES.
6169
6170 032056 012702 000017 ;MOV #17,R2 ;SET LOOP COUNT.
6171 032062 016704 150150 ;MOV CSRA,R4 ;GET CSR ADDRESS.
6172 032066 110214 ;MOV8 R2,(R4) ;WRITE COUNT TO CSR.
6173 032070 111401 ;MOVB (R4),R1 ;READ BACK THE CONTENTS OF THE CSR
6174 032072 042701 177760 ;BIC #177760,R1 ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
6175 032076 020102 ;CMP R1,R2 ;CHECK FOR CORRECT DATA WRITTEN/READ.
6176 032100 001412 ;BEQ 4 ;IS EXPECTED DATA BAD? NO, SKIP ERROR REPORT.
6177 ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE 0 (D)."
6178 032102 012767 016352 151660 ;MOV #ER1601,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
6179 032110 005003 ;CLR R3 ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
6180 032112 005005 ;CLR R5 ;CAUSE REPORT OF LINE 0.
6181 032114 104460 ;ERROR ;>>>> ERROR # 1802 <<<<
6182 ;TRAP C#ERROR
6183
6184 ;*
6185 ;* EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
6186
6187 032116 032767 000100 150074 ;BIT #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
6188 032124 001456 ;BEQ 60 ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
6189 ;DURING THE SOFTWARE QUESTIONS.
```

```

6185 032126 005302      44:      DEC      R2      ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
6186 032130 002356      BGE      24      ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
6187
6188      ;*
6189      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6190      ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6191      ; EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6192      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1803 - 1805 <<<<.
6193 032132 005267 151626      INC      ERRNBR      ;SET THE ERROR NUMBER TO 1803.
6194 032136 012703 177777      MOV      #-1,R3      ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6195 032142 012704 000002      MOV      #2,R4      ;INDICATE R/W ACCESS, CLEAR FIRST.
6196 032146 004767 166660      JSR      PC,REGTST      ;WRITE AND VERIFY DATA PATTERNS.
6197
6198      ;*
6199      ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6200      ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6201 032152 005767 150122      TST      EXOERR      ;IS THE "EXIT ON ERROR" FLAG SET ?
6202 032156 001041      BNE      604      ;EXIT IF IT IS.
6203
6204      ;*
6205      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6206      ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6207      ; EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6208      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1806 - 1808 <<<<.
6209      ;*
6210 032160 012767 003416 151576      MOV      #1806.,ERRNBR      ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6211 032166 005403      NEG      R3      ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6212 032170 004767 166636      JSR      PC,REGTST      ;WRITE AND VERIFY DATA PATTERNS.
6213
6214      ;*
6215      ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6216      ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6217 032174 005767 150100      TST      EXOERR      ;IS THE "EXIT ON ERROR" FLAG SET ?
6218 032200 001030      BNE      604      ;EXIT IF IT IS.
6219
6220      ;*
6221      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6222      ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6223      ; EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6224      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1809 - 1811 <<<<.
6225 032202 012767 003421 151554      MOV      #1809.,ERRNBR      ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6226 032210 005403      NEG      R3      ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6227 032212 005404      NEG      R4      ;INDICATE R/W ACCESS, SET FIRST.
6228 032214 004767 166612      JSR      PC,REGTST      ;WRITE AND VERIFY DATA PATTERNS.
6229
6230      ;*
6231      ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6232      ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6233 032220 005767 150054      TST      EXOERR      ;IS THE "EXIT ON ERROR" FLAG SET ?
6234 032224 001016      BNE      604      ;EXIT IF IT IS.
6235
6236      ;*
6237      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6238      ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6239      ; EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6240      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1812 - 1814 <<<<.
6241 032226 012767 003424 151530      MOV      #1812.,ERRNBR      ;SET UP ERROR NUMBER FOR REGTST ROUTINE.

```

```

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

```

L10046:

TRAP C\$ETST

```

6259 .SBTTL  HARDWARE TEST          - REGBRM -
6260 ;* *****
6261 ;* - DEVICE REGISTER BYTE ACCESS READ/MODIFY/WRITE TEST -
6262 ;*
6263 ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
6264 ;* CORRECTLY USING BYTE ACCESSES IN READ/MODIFY/WRITE MODE.
6265 ;*
6266 ;* -- *****
6267
6268 032270          BGNTST
6269 032270          SETPRI #PRI05          ;ALLOW THE LTC TO INTERRUPT.
6270 032270          012700 000240          T18::
6271 032274          104441          MOV #PRI05,R0
6272 032274          000022          TRAP C#SPRI
6273 032276          012767 000022 150014  TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6274 032304          012767 177777 147770  MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (19)
6275 032312          012767 000001 151442  MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
6276 032320          012767 003555 151436  MOV #1,ERRTYP ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6277 032326          012767 012467 151432  MOV #1901.,ERRNBR ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6278 032334          005067 150116          MOV #EM1901,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6279 032340          012700 002460          CLR ERSNRF ;CLEAR THE ERROR SUMMARY FLAGS.
6280 032344          004767 165410          MOV #ERCNTB,R0
6281 032350          005067 147724          JSR PC,CLR16W ;CLEAR THE ERROR COUNTER TABLE.
6282 ;*
6283 ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6284 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6285 ; THIS SUBROUTINE REPORTS ERRORS >>>> 1901 <<<<.
6286 ;*
6287 032354          004767 166732          JSR PC,RESETT ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6288 032360          103402          BCS .+6 ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
6289 032362          000167 000136          JMP 601 ;YES, EXIT THE TEST.
6290 032366          012767 003557 151370  MOV #1903.,ERRNBR ;SET THE ERROR REPORT NUMBER TO 1903.
6291 ;*
6292 ; THE READ/MODIFY/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR IS NOT
6293 ; TESTED SINCE THIS IS AN ILLEGAL FORM OF ACCESS TO THIS REGISTER.
6294 ;*
6295 ;*
6296 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6297 ; REGISTERS ON ALL ACTIVE LINES. USE READ/MODIFY/WRITE ACCESSES. BEFORE
6298 ; WRITING EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6299 ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1903 - 1905 <<<<.
6300 ;*
6301 032374          005267 151364          INC ERRNBR ;SET THE ERROR NUMBER TO 1903.
6302 032400          012703 177777          MOV #-1,R3 ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6303 032404          012704 000001          MOV #1,R4 ;INDICATE R/M/W ACCESS, CLEAR FIRST.
6304 032410          004767 166416          JSR PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6305 ;*
6306 ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6307 ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6308 ;*
6309 032414          005767 147660          TST EXOERR ;IS THE "EXIT ON ERROR" FLAG SET ?
6310 032420          001041          BNE 601 ;EXIT IF IT IS.
6311 ;*
6312 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL

```

```

6313 ; REGISTERS ON ALL ACTIVE LINES. USE READ/MODIFY/WRITE ACCESSES. BEFORE
6314 ; WRITING EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6315 ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1906 - 1908 <<<<.
6316 ;
6317 032422 012767 003562 151334 ; MOV #1906.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6318 032430 005403 ; NEG R3 ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6319 032432 004767 166374 ; JSR PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6320 ;
6321 ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6322 ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6323 ;
6324 032436 005767 147636 ; TST EXOERR ;IS THE "EXIT ON ERROR" FLAG SET ?
6325 032442 001030 ; BNE 604 ;EXIT IF IT IS.
6326 ;
6327 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6328 ; REGISTERS ON ALL ACTIVE LINES. USE READ/MODIFY/WRITE ACCESSES. BEFORE
6329 ; WRITING EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6330 ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1909 - 1911 <<<<.
6331 ;
6332 032444 012767 003565 151312 ; MOV #1909.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6333 032452 005403 ; NEG R3 ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6334 032454 005404 ; NEG R4 ;INDICATE R/M/W ACCESS. SET FIRST.
6335 032456 004767 166350 ; JSR PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6336 ;
6337 ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6338 ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6339 ;
6340 032462 005767 147612 ; TST EXOERR ;IS THE "EXIT ON ERROR" FLAG SET ?
6341 032466 001016 ; BNE 604 ;EXIT IF IT IS.
6342 ;
6343 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6344 ; REGISTERS ON ALL ACTIVE LINES. USE READ/MODIFY/WRITE ACCESSES. BEFORE
6345 ; WRITING EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6346 ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1912 - 1914 <<<<.
6347 ;
6348 032470 012767 003570 151266 ; MOV #1912.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6349 032476 005403 ; NEG R3 ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6350 032500 004767 166326 ; JSR PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6351 ;
6352 ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6353 ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6354 ;
6355 032504 005767 147570 ; TST EXOERR ;IS THE "EXIT ON ERROR" FLAG SET ?
6356 032510 001005 ; BNE 604 ;EXIT IF IT IS.
6357 ;
6358 ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6359 ; THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>> ERROR # 1915 <<<<
6360 ;
6361 032512 012767 003573 151244 ; MOV #1915.,ERRNBR ;SET UP ERROR NUMBER FOR NEXT RTN.
6362 032520 004767 166540 ; JSR PC,REPSMR ;REPORT ERROR SUMMARY IF NECESSARY.
6363 032524 005067 147552 604: CLR CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
6364 032530 ;
6365 032530 104401 ;

```

L10047:

TRAP C#ETST

```

6366 .SBTTL  HARDWARE TEST          - IDBIT -
6367 ;* *****
6368 ;* - DEVICE REGISTER ID BIT TEST -
6369 ;*
6370 ;* THIS TEST VERIFIES THAT THE DUT STAT REGISTER ID BIT READS AS SET.
6371 ;*
6372 ;* *****
6373
6374 032532          BGNTST
6375 032532          T19::
        032532          SETPRI #PRI05          ;ALLOW THE LTC TO INTERRUPT.
        032536 012700 000240          MOV #PRI05,R0
        032536 104441          TRAP C#SPRI
6376 000023          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6377 032540 012767 000023 147552          MOV #TNUM,TSTNUM          ;SET UP THE TEST NUMBER. (20)
6378 032546 012767 177777 147526          MOV #-1,CTRLCF          ;INDICATE THAT WE ARE IN A TEST.
6379 032554 012767 000001 151200          MOV #1,ERRTYP          ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6380 032562 012767 003721 151174          MOV #2001,ERRNBR          ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6381 032570 012767 012552 151170          MOV #EM2001,ERRMSG          ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6382
6383 ;*
6384 ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6385 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6386 ; THIS SUBROUTINE REPORTS ERRORS >>>> 2001 <<<<.
6387
6387 032576 004767 166510          JSR PC,RESETT          ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6388 032602 103016          BCC 60$          ;FATAL RESET ERROR? YES, EXIT THE TEST.
6389
6390 ;*
6391 ; READ THE STAT REGISTER ID BIT AND VERIFY THAT IT IS CLEAR.
6392
6392 032604 017701 147434          MOV #FSLSA,R1          ;READ THE STAT REGISTER CONTENTS.
6393 032610 032701 000400          BIT #BIT8,R1          ;CHECK THE ID BIT.
6394 032614 001011          BNE 60$          ;ID BIT SET? YES, EXIT THE TEST.
6395 032616 012767 003722 151140          MOV #2002,ERRNBR          ;NO, SET THE ERROR REPORT NUMBER TO 2002.
6396 032624 012701 012622          MOV #EM2002,R1          ;GET THE PROPER ERROR MESSAGE.
6397 032630 012767 016072 151132          MOV #ER0503,ERRBLK          ;SELECT THE PROPER ERROR REPORT ROUTINE.
6398 032636          ERROR          ;ERROR NUMBER >>>> 2002 <<<<
        032636 104460          TRAP C#ERROR
6399 032640 005067 147436          60$: CLR CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
6400 032644          ENDTST
        032644          L10050:
        032644 104401          TRAP C#ETST

```

```

6402 .SBTTL HARDWARE TEST - TXENBI-
6403 ;* *****
6404 ;* - TX_ENABLE (INACTIVE) TEST -
6405 ;* THIS TEST VERIFIES THAT WHEN THE LINE UNDER TEST'S TX_ENABLE BIT IS
6406 ;* CLEAR, TRANSMISSION WILL NOT TAKE PLACE ON THAT LINE.
6407 ;* THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, AND ON ALL ACTIVE LINES.
6408 ;*
6409 ;* *****
6410
6411 032646 BGNTST
6412 032646 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T20::
6413 032646 012700 000240 MOV #PRI05,R0
6414 032652 104441 TRAP C#SPRI
6415 032652 000024 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6416 032654 012767 000024 147436 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (23)
6417 032662 012767 177777 147412 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
6418 032670 012767 000001 151064 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6419 032676 012767 004375 151060 MOV #2301,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
6420 032704 012767 012675 151054 MOV #EM2301,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
6421 032712 012767 017156 151050 MOV #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6422
6423 ;*
6424 ;* RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
6425 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6426 ;* THIS SUBROUTINE REPORTS ERROR >>>> 2301 <<<<.
6427 ;*
6428 JSR PC,CLNRST ;RESET THE DMU-11, REPORT ANY ERRORS FOUND.
6429 BCC 608 ;RESET FAILURE?, ABORT THIS TEST.
6430
6431 ;*
6432 ;* SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
6433 ;* SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
6434 ;* 2 STOP BITS.
6435 ;* ENABLE TRANSMITTERS ON ALL LINES.
6436 ;*
6437 MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
6438 MOV #200,R0 ;PASS THE LNCTRL CONTENTS.
6439 JSR PC,WTMLNC ;INITIALISE THE LNCTRL REGISTERS.
6440 MOV #177670,R0 ;PASS THE LPR CONTENTS.
6441 JSR PC,WTMLPR ;INITIALISE THE LPR REGISTERS ON ALL LINES.
6442 MOV #10,R4 ;PASS DELAY TIME OF 10 MILLI-SECONDS.
6443 JSR PC,DELAY ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
6444 MOV #MAPLNS,R5 ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
6445 JSR PC,TXENBL ;ENABLE TRANSMITTERS ON ALL LINES.
6446
6447 ;*
6448 ;* TEST ALL ACTIVE LINES INDIVIDUALLY.
6449 ;* DISABLE TRANSMISSION ON EACH ACTIVE LINE.
6450 ;*
6451 MOV #1,R3 ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
6452 CLR R4 ;CLEAR THE LINE NUMBER COUNTER.
6453 MOV #2302,ERRNBR ;SET THE ERROR NUMBER TO 2302.
6454 BIT R3,ACTLNS ;CHECK IF THE LINE IS ACTIVE.
6455 BEQ 61 ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
6456
6457 ;*
6458 ;* CLEAR THE TX_ENABLE BIT IN TBUFFAD2 REGISTER.
6459 ;* SELECT THE LINE UNDER TEST.
6460 ;* VERIFY IT IS CLEAR, REPORT ERROR IF SET.
6461 ;*

```

C14

```

6456 033014 010305      MOV    R3,R5      ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6457 033016 004767 167516 JSR    PC,TXDSBL    ;DISABLE TRANSMISSION ON THE LINE UNDER TEST.
6458 033022 010477 147210 MOV    R4,BCSRA     ;SELECT THE LINE CURRENTLY UNDER TEST.
6459 033026 005777 147220 TST    BTXAD2A      ;VERIFY THE TX_ENABLE BIT IS SET.
6460 033032 100433      BMI     41      ;GO REPORT ERROR IF TX_ENABLE BIT SET.
6461
6462      ;*
6463      ; WRITE DATA BYTE (ASCII <LF>) TO THE OUTPUT FIFO.
6464      ; WAIT FOR A TX_ACTION TO BE RETURNED, REPORT ERROR IF A TX_ACTION
6465      ; IS FOUND BEFORE TIME-OUT OCCURS.
6466 033034 012767 004377 150722 MOV    #2303,ERRNBR ;SET ERROR NUMBER TO 2303.
6467 033042 112777 000012 147174 MOVB   #12,BFDATA   ;WRITE THE DATA BYTE TO THE DUT'S OUTPUT FIFO.
6468 033050 012701 170003      MOV    #170003,R1    ;TEST BIT 15, TIMEOUT OF 3 MILLI SECS.
6469 033054 016702 147156      MOV    CSRA,R2     ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6470 033060 004767 170064      JSR    PC,WAIBIS    ;WAIT FOR TX_ACTION TO COME BACK.
6471 033064 103416      BCS     41      ;GO REPORT ERROR IF A TX-ACTION FOUND.
6472
6473      ;*
6474      ; WAIT FOR THE DATA TO APPEAR IN THE FIFO, REPORT ERROR IF DATA FOUND.
6475 033066 005267 150672      INC     ERRNBR      ;SET ERROR NUMBER TO 2304.
6476 033072 012701 070012      MOV    #70012,R1     ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
6477 033076 016702 147134      MOV    CSRA,R2     ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6478 033102 004767 170042      JSR    PC,WAIBIS    ;WAIT FOR RX_DATA AVAILABLE TO SET.
6479 033106 103405      BCS     41      ;REPORT ERROR IF DATA RECEIVED IN THE FIFO.
6480 033110 005267 150650      INC     ERRNBR      ;SET ERROR NUMBER TO 2305.
6481 033114 017702 147120      MOV    BRBUFA,R2    ;READ THE DATA FROM THE FIFO.
6482 033120 100010      BPL     61      ;SKIP ERROR REPORT IF DATA ISN'T THERE.
6483
6484 033122 010401      41:    MOV    R4,R1      ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6485 033124 012702 012742      MOV    #EM2302,R2    ;PASS THE MESSAGE TO BE REPORTED.
6486
6487 033130      ERROR      ; "TX_ENABLE BIT BAD ON LINE: NN".
6488 033130 104460      ; >>>> ERROR <<<<<
6489
6489 033132 032767 000100 147060 BIT     #BIT06,OPTION ;EXIT THE TEST, WITH THE TEST FAILED MESSAGE.
6490 033140 001406      BEQ     601      ;IF EXTENDED ERROR REPORTING HAS NOT BEEN
6491
6492
6493
6494
6495      ;*
6496 033142 000241      ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6497 033144 006103      ;*
6498 033146 005204      61:    CLC          ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
6499 033150 020427 000020      ROL     R3      ;SHIFT THE BIT MAP FOR THE NEXT LINE.
6500 033154 002711      INC     R4      ;INCREMENT THE LINE NUMBER COUNTER.
6501
6502 033156 005067 147120      CMP    R4,#NUMLNS ;HAVE ALL THE LINES BEEN TESTED?
6503 033162      BLT     21      ;NO, BRANCH TO TEST THE NEXT LINE.
6503 033162      601:    CLR     CTRLCF    ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6503 033162 104401      ENDTST

```

L10051: TRAP C#ETST

```

6505 .SBTTL HARDWARE TEST - TXENBA-
6506 ;* *****
6507 ;* - TX_ENABLE (ACTIVE) TEST -
6508 ;* THIS TEST VERIFIES THAT WHEN THE TX_ENABLE BIT IS SET IN THE APPROPRIATE
6509 ;* LINE REGISTER, TRANSMISSION WILL TAKE PLACE ON THAT LINE.
6510 ;* THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, AND ON ALL ACTIVE LINES.
6511 ;*
6512 ;* *****
6513 ;*
6514 033164 BGNTST
        033164
6515 033164 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T21::
        033164 012700 000240 MOV #PRI05,R0
        033170 104441 TRAP C#SPRI
6516 000025 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6517 033172 012767 000025 147120 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (24)
6518 033200 012767 177777 147074 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
6519 033206 012767 000001 150546 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6520 033214 012767 004541 150542 MOV #2401,,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
6521 033222 012767 013000 150536 MOV #EM2401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTABL.
6522 033230 012767 017156 150532 MOV #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6523 ;*
6524 ;* RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
6525 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6526 ;* THIS SUBROUTINE REPORTS ERROR >>>> 2401 <<<<.
6527 ;*
6528 033236 004767 164474 JSR PC,CLNRST ;RESET THE DMU-11, REPORT ANY ERRORS FOUND.
6529 033242 103133 BCC 601 ;RESET FAILURE?, ABORT THIS TEST.
6530 ;*
6531 ;* SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
6532 ;* SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
6533 ;* 2 STOP BITS.
6534 ;* DISABLE TRANSMITTERS ON ALL LINES.
6535 ;*
6536 033244 016705 146754 MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
6537 033250 012700 000200 MOV #200,R0 ;PASS THE LNCTRL CONTENTS.
6538 033254 004767 170152 JSR PC,WTMLNC ;INITIALISE THE LNCTRL REGISTERS.
6539 033260 012700 177670 MOV #177670,R0 ;PASS THE LPR CONTENTS.
6540 033264 004767 170172 JSR PC,WTMLPR ;INITIALISE THE LPR REGISTERS ON ALL LINES.
6541 033270 012704 000012 MOV #10,,R4 ;PASS DELAY TIME OF 10 MILLI-SECONDS.
6542 033274 004767 164560 JSR PC,DELAY ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
6543 033300 012705 177777 MOV #MAPLNS,R5 ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
6544 033304 004767 167230 JSR PC,TXDSBL ;DISABLE TRANSMITTERS ON ALL LINES.
6545 ;*
6546 ;* TEST ALL ACTIVE LINES INDIVIDUALLY.
6547 ;* ENABLE TRANSMISSION ON EACH ACTIVE LINE.
6548 ;*
6549 033310 012703 000001 MOV #1,R3 ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
6550 033314 005004 CLR R4 ;CLEAR THE LINE NUMBER COUNTER.
6551 033316 012767 004542 150440 24: MOV #2402,,ERRNBR ;SET THE ERROR NUMBER TO 2402.
6552 033324 030367 146674 BIT R3,ACTLNS ;CHECK IF THE LINE IS ACTIVE.
6553 033330 001467 BEQ 81 ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
6554 ;*
6555 ;* SELECT THE LINE UNDER TEST.
6556 ;* SET THE TX_ENABLE BIT IN TBUFAD2 REGISTER.
6557 ;* VERIFY IT IS SET, REPORT ERROR IF CLEAR.
6558 ;*

```

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

6615 033532 005067 146544

601:

CLR CTRLCF
ENDTST

;INDICATE THAT WE ARE NOT WITHIN A TEST.

6616 033536
033536
033536

104401

L10052:

TRAP C#ETST

```
6618 .SBTTL  HARDWARE TEST - INTA -
6619 ;* *****
6620 ;* - INTERRUPT TEST -
6621 ;* THIS TEST VERIFIES THAT THE DEVICE UNDER TEST (DUT) WILL GENERATE
6622 ;* RECEPTION AND TRANSMISSION INTERRUPTS CORRECTLY. THIS TEST DOES
6623 ;* NOT DEPEND ON THE USE OF THE SERIAL LINE TRANSMISSION OR RECEPTION
6624 ;* CAPABILITIES OF THE DUT. THE LINES ARE PUT IN INTERNAL LOOPBACK
6625 ;* TO MINIMIZE ANY EXTERNAL EFFECTS THAT COULD BE CAUSED ON DEVICES
6626 ;* ATTACHED TO THE SERIAL LINES.
6627 ;*
6628 ;* *****
6629 ;*
6630 BGNTST
6631 033540          T22::
033540          SETPRI  #PRI05          ;ALLOW THE LTC TO INTERRUPT.
033540          MOV      #PRI05,R0
033544          TRAP    C$SPRI
6632          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6633 033546 012700 000240          MOV      #TNUM,TSTNUM          ;SET UP THE TEST NUMBER. (26)
6634 033554 012767 177777 146544  MOV      #-1,CTRLCF          ;INDICATE THAT WE ARE IN A TEST.
6635 033562 012767 000001 150172  MOV      #1,ERRTYP          ;SET ERROR FATAL ERROR TYPE IN ERROR TABLE.
6636 033570 012767 003101 150166  MOV      #1601,,ERRNBR          ;SET FIRST ERROR REPORT NUMBER IN ERROR TABLE.
6637 033576 012767 013043 150162  MOV      #EM2601,,ERRMSG          ;SET TEST ERROR MESSAGE IN ERROR TABLE.
6638 ;*
6639 ;* RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6640 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6641 ;* THIS SUBROUTINE REPORTS ERRORS FROM >>>> 2601 THRU 2602 <<<<.
6642 ;*
6643 033604 004767 165502          JSR      PC,RESETT          ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6644 033610 103402          BCS      2$          ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
6645 033612 000167 001044          JMP      60$          ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
6646 033616 012767 005053 150140 2$: MOV      #2603,,ERRNBR          ;SET THE ERROR REPORT NUMBER TO 2603.
6647 ;*
6648 ;* ENABLE TRANSMITTERS ON ALL LINES.
6649 ;*
6650 033624 012705 177777          4$: MOV      #MAPLNS,R5          ;PASS ACTIVE LINE BIT MAP.
6651 033630 004767 167000          JSR      PC,TXENBL          ;ENABLE TRANSMISSION ON ALL LINES.
6652 ;*
6653 ;* TEST RECEPTION INTERRUPTS.
6654 ;* SET UP FOR RX AND TX INTERRUPTS:
6655 ;* RX INTERRUPT SERVICE ROUTINE INPUTS A CHAR AND COUNTS THE INTERRUPT.
6656 ;* TX INTERRUPT SERVICE ROUTINE COUNTS TX INTERRUPTS.
6657 ;*
6658 033634 005067 146450          CLR      RXINTC          ;CLEAR THE RX INTERRUPT COUNTER.
6659 033640 005067 146446          CLR      RXINTF          ;CLEAR THE RX INTERRUPT FLAGS.
6660 033644 005067 146452          CLR      TXINTC          ;CLEAR THE TX INTERRUPT COUNTER.
6661 033650 012767 002722 146420  MOV      #BUFBAS,BUFPTR          ;LOAD THE BUFFER PTR WITH THE BUFFER BASE ADR.
6662 033656          SETVEC  RXVECA,#RXINPT,#PRI06 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
033656          MOV      #PRI06,-(SP)
033662 012746 023716          MOV      #RXINPT,-(SP)
033666 016746 146334          MOV      RXVECA,-(SP)
033672 012746 000003          MOV      #3,-(SP)
033676 104437          TRAP    C$SVEC
033700 062706 000010          ADD      #10,SP
6663 033704          SETVEC  TXVECA,#CACHTX,#PRI06 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
033704          MOV      #PRI06,-(SP)
033710 012746 023540          MOV      #CACHTX,-(SP)
```

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

```

6711      ; VERIFY THAT NO TX INTERRUPTS HAVE BEEN GENERATED SO FAR IN THIS TEST.
6712      ;-
6713 034062 016702 146234      101:  MOV    TXINTC,R2      ;LOAD # OF TX INTERRUPTS FOR ER0504 RTN.
6714 034066 001414      BEQ     121      ;SKIP ERROR IF NO TX INTERRUPTS.
6715      ;REPORT "TX INTERRUPTS(S) RECEIVED WITH TX INTERRUPTS DISABLED."
6716 034070 012701 007521      MOV    #EM0526,R1      ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
6717 034074      ERRDF  2604,EM2601,ER0504;      >>>> ERROR #2604 <<<<.
        034074 104455      TRAP    C#ERCF
        034076 005054      .WORD   2604
        034100 013043      .WORD   EM26C1
        034102 016130      .WORD   ER0504

6718
6719
6720      ;*
6721      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
6722      ;-
6723 034104 032767 000100 146106      BIT    #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
6724 034112 001002      BNE     .+6      ;
6725 034114 000167 000520      JMP     341      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
6726
6727      ;*
6728      ; CLEAN OUT THE INTERRUPT VECTORS USED IN THIS TEST.
6729      ;-
6730 034120      121:  SETPRI  #PRI06      ;DISABLE DEVICE INTERRUPTS.
        034120 012700 000300      MOV    #PRI06,R0
        034124 104441      TRAP    C#SPRI
6731 034126      CLRVEC  RXVECA      ;RETURN RX INT VECTOR TO UNUSED POOL.
        034126 016700 146074      MOV    RXVECA,R0
        034132 104436      TRAP    C#CVEC
6732 034134      CLRVEC  TXVECA      ;RETURN TX INT VECTOR TO UNUSED POOL.
        034134 016700 146070      MOV    TXVECA,R0
        034140 104436      TRAP    C#CVEC
6733
6734      ;*
6735      ; TEST TRANSMISSION INTERRUPTS.
6736      ; SET UP FOR RX AND TX INTERRUPTS:
6737      ; RX INTERRUPT SERVICE ROUTINE COUNTS RX INTERRUPTS.
6738      ; TX INTERRUPT SERVICE ROUTINE COUNTS THE INTERRUPT AND SETS FLAGS.
6739      ;-
6739 034142 005067 146142      CLR    RXINTC      ;CLEAR THE RX INTERRUPT COUNTER.
6740 034146 005067 146150      CLR    TXINTC      ;CLEAR THE TX INTERRUPT COUNTER.
6741 034152 005067 146146      CLR    TXINTF      ;CLEAR THE RX INTERRUPT FLAGS.
6742 034156      SETVEC  RXVECA,#CACHRX,#PRI06 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
        034156 012746 000300      MOV    #PRI06,-(SP)
        034162 012746 023512      MOV    #CACHRX,-(SP)
        034166 016746 146034      MOV    RXVECA,-(SP)
        034172 012746 000003      MOV    #3,-(SP)
        034176 104437      TRAP    C#SVEC
        034200 062706 000010      ADD    #10,SP
6743 034204      SETVEC  TXVECA,#TXINTR,#PRI06 ;SET UP INT VECTOR TO TX INT ROUTINE.
        034204 012746 000300      MOV    #PRI06,-(SP)
        034210 012746 024026      MOV    #TXINTR,-(SP)
        034214 016746 146010      MOV    TXVECA,-(SP)
        034220 012746 000003      MOV    #3,-(SP)
        034224 104437      TRAP    C#SVEC
        034226 062706 000010      ADD    #10,SP
6744 034232      SETPRI  #PRI04      ;ALLOW DEVICE INTERRUPTS.
        034232 012700 000200      MOV    #PRI04,R0

```

```
034236 104441                                TRAP    C$SPRI
6745
6746
6747
6748 034240 012705 000022
6749 034244 012701 000144
6750 034250 012702 100000
6751 034254 016704 145756
6752 034260 012703 100000
6753 034264 004767 163744
6754 034270 103026
6755 034272 005003
6756 034274 004767 163734
6757 034300 103005
6758 034302 005305
6759 034304 001365
6760
6761 034306 012701 013476
6762 034312 000402
6763 034314 012701 013572
6764 034320
      034320 104455
      034322 005055
      034324 013437
      034326 016072
6765
6766
6767
6768
6769 034330 032767 000100 145662
6770 034336 001540
6771 034340 004767 166424
6772 034344 000430
6773
6774
6775
6776 034346 004767 166416
6777 034352 012704 000062
6778 034356 004767 163476
6779 034362 005767 145734
6780 034366 001417
6781 034370 012701 013476
6782 034374 005767 145724
6783 034400 100002
6784 034402 012701 013643
6785
6786 034406
      034406 104455
      034410 005056
      034412 013437
      034414 016072
6787
6788
6789
6790
6791 034416 032767 000100 145574
6792 034424 001500

;+
; VERIFY THAT THE TX_ACTION BIT IS CLEAR.
;-
      MOV    #18.,R5          ;INITIALIZE THE LOOP COUNTER.
      MOV    #100.,R1         ;SET 100 MS TIME-OUT.
      MOV    #BIT15,R2        ;SELECT TX ACTION BIT TO TEST.
      MOV    CSRA,R4           ;PASS OUT CSR AS THE WORD TO TEST.
14+:   MOV    #BIT15,R3        ;WAIT FOR TX_ACTION TO BE SET.
      JSR    PC,MSLOOP         ;WAIT UP TO 100 MS FOR TX_ACTION SET.
      BCC    20:              ;IF TIME-OUT, CONSIDER TX_ACTION CLEAR.
      CLR    R3               ;NOW, WAIT FOR TX_ACTION CLEAR.
      JSR    PC,MSLOOP         ;WAIT UP TO 100 MS FOR TX_ACTION CLEAR.
      BCC    16:              ;IF TIME-OUT, REPORT TX_ACTION WON'T CLEAR.
      DEC    R5               ;DECREMENT THE TX_ACTION SET COUNTER.
      BNE    14:              ;LOOP IF NOT TOO MANY TX_ACTIONS FOUND.
;REPORT "TX_ACTION SET REPEATEDLY AFTER RESET, NO DATA SENT."
      MOV    #EM2607,R1        ;SELECT ERROR MESSAGE.
      BR     18:              ;GO TO REPORT THE ERROR.
16+:   MOV    #EM2608,R1        ;SELECT TX_ACTION STUCK SET MSG.
18+:   ERDF    2605,EM2606,ER0503; >>>>> ERROR #2605 <<<<<.
                                           TRAP    C$ERDF
                                           .WORD    2605
                                           .WORD    EM2606
                                           .WORD    ER0503

;+
; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
;-
      BIT    #BIT06,OPTION     ;EXIT WITH TEST FAILURE MESSAGE IF
      BEQ    34:              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
      JSR    PC,TXIE1          ;ENABLE TX INTERRUPTS FOR THE TX_INT TESTING.
      BR     24:              ;GO TO TEST WITH TX_ACTION SET.

;+
; VERIFY THAT NO INTERRUPTS OCCUR WITH TX_ACTION CLEAR.
;-
20+:   JSR    PC,TXIE1          ;ENABLE TX INTERRUPTS.
      MOV    #50.,R4           ;PASS 50 MS TIME TO THE DELAY ROUTINE.
      JSR    PC,DELAY          ;DELAY 50 MILLI-SECONDS TO ALLOW INTS TO OCCUR.
      TST    TXINTC            ;TEST THE TX INTERRUPT COUNT.
      BEQ    24:              ;SKIP THE ERROR IF NO TX INTERRUPTS.
      MOV    #EM2607,R1        ;SELECT MESSAGE IN CASE TX INT FLAG CLEAR.
      TST    TXINTF            ;TEST THE TX INTERRUPT FLAGS.
      BPL    22:              ;GO REPORT ERROR IF TX FLAG IS CLEAR.
      MOV    #EM2609,R1        ;TX FLAG IS SET, SELECT PROPER ERROR MESSAGE.
;REPORT "TRANSMIT INTERRUPT TEST ERROR:..."
22+:   ERDF    2606,EM2606,ER0503; >>>>> ERROR #2606 <<<<<.
                                           TRAP    C$ERDF
                                           .WORD    2606
                                           .WORD    EM2606
                                           .WORD    ER0503

;+
; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
;-
      BIT    #BIT06,OPTION     ;EXIT WITH TEST FAILURE MESSAGE IF
      BEQ    32:              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
```

```

6793
6794
6795
6796
6797 034426 005067 145670
6798 034432 005067 145666
6799
6800
6801
6802 034436 012705 177777
6803 034442 012700 000200
6804 034446 004767 166760
6805 034452 012700 156430
6806 034456 004767 167000
6807
6808
6809
6810 034462 016701 145616
6811 034466 005002
6812 034470 010177 145542
6813
6814 034474 112777 000000 145542
6815 034502 005201
6816 034504 005202
6817 034506 020227 000020
6818 034512 002766
6819
6820
6821
6822 034514 012704 000372
6823 034520 004767 163334
6824
6825
6826
6827 034524 005767 145572
6828 034530 001010
6829
6830
6831
6832 034532 012701 013722
6833 034536 005777 145474
6834 034542 100410
6835 034544 012701 014014
6836 034550 000405
6837
6838
6839
6840 034552 005767 145546
6841 034556 100012
6842 034560 012701 013643
6843
6844
6845
6846 034564
      034564 104455
      034566 005057
      034570 013437

;
; PREPARE TX INTERRUPT COUNTER AND FLAGS.
;
24: CLR TXINTC ;CLEAR THE TX INTERRUPT COUNT.
    CLR TXINTF ;CLEAR THE TX INTERRUPT FLAGS.
;
; SET UP LINE PARAMETERS FOR TRANSMISSION.
;
    MOV #MAPLNS,R5 ;PASS ACTIVE LINES BIT MAP.
    MOV #200,R0 ;PASS INERT STATE, INTERNAL LOOPBACK.
    JSR PC,WTWLCN ;DISABLE RECEPTION AND DMA, ETC. ON OUT.
    MOV #156430,R0 ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
    JSR PC,WTWLPR ;WRITE TO ALL LPR REGISTERS.
;
; SEND A NULL CHAR TO EACH LINE.
;
    MOV IESTAT,R1 ;SET UP THE STATE OF THE INTERRUPT ENABLE BITS.
    CLR R2 ;CLEAR THE LINE COUNTER.
25: MOV R1,BCSRA ;SET UP THE LINE NUMBER AND INTERRUPT ENABLE
    ;BITS IN THE CSR.
    MOVB #0,BFDATA ;SEND A NULL CHARACTER TO THE OUTPUT FIFO.
    INC R1 ;NEXT CSR CONTENTS.
    INC R2 ;NEXT LINE.
    CMP R2,#NUMLNS ;IF ALL LINES HAVE NOT BEEN SERVICED THEN
    BLT 25 ;BRANCH.
;
; DELAY 250 MILLI-SECONDS TO ALLOW INTERRUPTS TO OCCUR.
;
    MOV #250,R4 ;SET UP FOR 250 MS DELAY.
    JSR PC,DELAY ;WAIT 250 MS.
;
; VERIFY THAT TX INTERRUPTS OCCURRED.
;
    TST TXINTC ;CHECK THE TX INTERRUPT COUNTER.
    BNE 26 ;SKIP THE FOLLOWING ERROR IF WE GOT TX INTS.
;
; DETERMINE THE REASON THAT WE RECEIVED NO INTERRUPTS.
;
    MOV #EM2610,R1 ;SET UP MSG IN CASE "TX_ACTION IS SET".
    TST BCSRA ;CHECK THE DUT CSR.
    BMI 28 ;GO TO REPORT ERROR IF TX_ACTION IS SET.
    MOV #EM2611,R1 ;SET UP "TX_ACTION NOT SET" MESSAGE.
    BR 28 ;GO AND REPORT THE ERROR.
;
; CHECK TO VERIFY THAT TX_ACTION WAS SET FOR EACH INTERRUPT.
;
26: TST TXINTF ;CHECK THE TX INTERRUPT FLAGS.
    BPL 30 ;SKIP ERROR IF TX_ACTION CLR FLAG IS CLEAR.
    MOV #EM2609,R1 ;SET UP TX INT WITH "TX_ACTION CLR" MSG.
;
; REPORT "TRANSMIT INTERRUPT TEST ERROR:...."
;
28: ERDF 2607,EM2606,ER0503, >>>>> ERROR #2607 <<<<<.
                                TRAP C:ERDF
                                .WORD 2607
                                .WORD EM2606

```

```

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

```

```

6877 .SBTTL  HARDWARE TEST          - BRLEVA -
6878 ;* *****
6879 ;* - BR LEVEL TEST B -
6880 ;* THIS TEST VERIFIES THAT THE DEVICE UNDER TEST (DUT) WILL GENERATE
6881 ;* RECEPTION AND TRANSMISSION INTERRUPTS AT THE CORRECT BR LEVEL.
6882 ;* THIS TEST DOES NOT DEPEND ON THE USE OF THE SERIAL LINE TRANSMISSION
6883 ;* OR RECEPTION CAPABILITIES OF THE DUT. THE LINES ARE PUT IN INTERNAL
6884 ;* LOOPBACK TO MINIMIZE ANY EXTERNAL EFFECTS THAT COULD BE CAUSED ON
6885 ;* DEVICES ATTACHED TO THE SERIAL LINES.
6886 ;*
6887 ;* *****
6888 ;*
6889 034676 BGNTST
6890 034676 SETPRI #PRI05          ;ALLOW LTC INTERRUPTS.      T23::
6891 034676 012700 000240          ;
6892 034702 104441          ;
6893 000027          ;
6894 034704 012767 000027 145406 TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6895 034712 012767 177777 145362 MOV #TNUM,TSTNUM          ;SET UP THE TEST NUMBER. (30)
6896 034720 012767 000001 147034 MOV #-1,CTRLCF          ;INDICATE THAT WE ARE IN A TEST.
6897 034726 012767 005671 147030 MOV #1,ERRTYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6898 034734 012767 014173 147024 MOV #3001.,ERRNBR          ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
6899 034742 005067 145510 MOV #EM3001,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
6900 ;* CLR ERSNRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
6901 ;*
6902 ;* RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6903 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6904 ;* THIS SUBROUTINE REPORTS ERRORS FROM >>>> 3001 THRU 3002 <<<<.
6905 ;*
6906 034746 004767 164340          ;
6907 034752 103402          ;
6908 034754 000167 000644          ;
6909 034760 012767 005673 146776 2$: JSR PC,RESETT          ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6910 ;* BCS 2$          ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
6911 ;* JMP 60$          ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
6912 ;* MOV #3003.,ERRNBR          ;SET THE ERROR REPORT NUMBER TO 3003.
6913 ;*
6914 ;* ENABLE TRANSMITTERS ON ALL LINES.
6915 ;*
6916 034766 012705 177777          ;
6917 034772 004767 165636          ;
6918 ;*
6919 ;* 4$: MOV #MAPLNS,R5          ;PASS ACTIVE LINE BIT MAP.
6920 ;* JSR PC,TXENBL          ;ENABLE TRANSMISSION ON ALL LINES.
6921 ;*
6922 ;* GENERATE A TRANSMISSION INTERRUPT REQUEST.
6923 ;* PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING INTS.
6924 ;*
6925 034776 012700 000340          ;
6926 034776 104441          ;
6927 035002 012746 000340          ;
6928 035004 012746 024026          ;
6929 035010 016746 145210          ;
6930 035014 012746 000003          ;
6931 035020 104437          ;
6932 035026 062706 000010          ;
6933 ;*
6934 ;* SET UP DUT FOR TRANSMISSION INTERRUPTS:
6935 ;* SET UP INTERNAL LOOPBACK.
6936 ;* SET UP LINE PARAMETERS FOR TRANSMISSION.
6937 ;*
6938 ;*
6939 ;*
6940 ;*
6941 ;*
6942 ;*
6943 ;*
6944 ;*
6945 ;*
6946 ;*
6947 ;*
6948 ;*
6949 ;*
6950 ;*
6951 ;*
6952 ;*
6953 ;*
6954 ;*
6955 ;*
6956 ;*
6957 ;*
6958 ;*
6959 ;*
6960 ;*
6961 ;*
6962 ;*
6963 ;*
6964 ;*
6965 ;*
6966 ;*
6967 ;*
6968 ;*
6969 ;*
6970 ;*
6971 ;*
6972 ;*
6973 ;*
6974 ;*
6975 ;*
6976 ;*
6977 ;*
6978 ;*
6979 ;*
6980 ;*
6981 ;*
6982 ;*
6983 ;*
6984 ;*
6985 ;*
6986 ;*
6987 ;*
6988 ;*
6989 ;*
6990 ;*
6991 ;*
6992 ;*
6993 ;*
6994 ;*
6995 ;*
6996 ;*
6997 ;*
6998 ;*
6999 ;*
7000 ;*
7001 ;*
7002 ;*
7003 ;*
7004 ;*
7005 ;*
7006 ;*
7007 ;*
7008 ;*
7009 ;*
7010 ;*
7011 ;*
7012 ;*
7013 ;*
7014 ;*
7015 ;*
7016 ;*
7017 ;*
7018 ;*
7019 ;*
7020 ;*
7021 ;*
7022 ;*
7023 ;*
7024 ;*
7025 ;*
7026 ;*
7027 ;*
7028 ;*
7029 ;*
7030 ;*
7031 ;*
7032 ;*
7033 ;*
7034 ;*
7035 ;*
7036 ;*
7037 ;*
7038 ;*
7039 ;*
7040 ;*
7041 ;*
7042 ;*
7043 ;*
7044 ;*
7045 ;*
7046 ;*
7047 ;*
7048 ;*
7049 ;*
7050 ;*
7051 ;*
7052 ;*
7053 ;*
7054 ;*
7055 ;*
7056 ;*
7057 ;*
7058 ;*
7059 ;*
7060 ;*
7061 ;*
7062 ;*
7063 ;*
7064 ;*
7065 ;*
7066 ;*
7067 ;*
7068 ;*
7069 ;*
7070 ;*
7071 ;*
7072 ;*
7073 ;*
7074 ;*
7075 ;*
7076 ;*
7077 ;*
7078 ;*
7079 ;*
7080 ;*
7081 ;*
7082 ;*
7083 ;*
7084 ;*
7085 ;*
7086 ;*
7087 ;*
7088 ;*
7089 ;*
7090 ;*
7091 ;*
7092 ;*
7093 ;*
7094 ;*
7095 ;*
7096 ;*
7097 ;*
7098 ;*
7099 ;*
7100 ;*
7101 ;*
7102 ;*
7103 ;*
7104 ;*
7105 ;*
7106 ;*
7107 ;*
7108 ;*
7109 ;*
7110 ;*
7111 ;*
7112 ;*
7113 ;*
7114 ;*
7115 ;*
7116 ;*
7117 ;*
7118 ;*
7119 ;*
7120 ;*
7121 ;*
7122 ;*
7123 ;*
7124 ;*
7125 ;*
7126 ;*
7127 ;*
7128 ;*
7129 ;*
7130 ;*
7131 ;*
7132 ;*
7133 ;*
7134 ;*
7135 ;*
7136 ;*
7137 ;*
7138 ;*
7139 ;*
7140 ;*
7141 ;*
7142 ;*
7143 ;*
7144 ;*
7145 ;*
7146 ;*
7147 ;*
7148 ;*
7149 ;*
7150 ;*
7151 ;*
7152 ;*
7153 ;*
7154 ;*
7155 ;*
7156 ;*
7157 ;*
7158 ;*
7159 ;*
7160 ;*
7161 ;*
7162 ;*
7163 ;*
7164 ;*
7165 ;*
7166 ;*
7167 ;*
7168 ;*
7169 ;*
7170 ;*
7171 ;*
7172 ;*
7173 ;*
7174 ;*
7175 ;*
7176 ;*
7177 ;*
7178 ;*
7179 ;*
7180 ;*
7181 ;*
7182 ;*
7183 ;*
7184 ;*
7185 ;*
7186 ;*
7187 ;*
7188 ;*
7189 ;*
7190 ;*
7191 ;*
7192 ;*
7193 ;*
7194 ;*
7195 ;*
7196 ;*
7197 ;*
7198 ;*
7199 ;*
7200 ;*
7201 ;*
7202 ;*
7203 ;*
7204 ;*
7205 ;*
7206 ;*
7207 ;*
7208 ;*
7209 ;*
7210 ;*
7211 ;*
7212 ;*
7213 ;*
7214 ;*
7215 ;*
7216 ;*
7217 ;*
7218 ;*
7219 ;*
7220 ;*
7221 ;*
7222 ;*
7223 ;*
7224 ;*
7225 ;*
7226 ;*
7227 ;*
7228 ;*
7229 ;*
7230 ;*
7231 ;*
7232 ;*
7233 ;*
7234 ;*
7235 ;*
7236 ;*
7237 ;*
7238 ;*
7239 ;*
7240 ;*
7241 ;*
7242 ;*
7243 ;*
7244 ;*
7245 ;*
7246 ;*
7247 ;*
7248 ;*
7249 ;*
7250 ;*
7251 ;*
7252 ;*
7253 ;*
7254 ;*
7255 ;*
7256 ;*
7257 ;*
7258 ;*
7259 ;*
7260 ;*
7261 ;*
7262 ;*
7263 ;*
7264 ;*
7265 ;*
7266 ;*
7267 ;*
7268 ;*
7269 ;*
7270 ;*
7271 ;*
7272 ;*
7273 ;*
7274 ;*
7275 ;*
7276 ;*
7277 ;*
7278 ;*
7279 ;*
7280 ;*
7281 ;*
7282 ;*
7283 ;*
7284 ;*
7285 ;*
7286 ;*
7287 ;*
7288 ;*
7289 ;*
7290 ;*
7291 ;*
7292 ;*
7293 ;*
7294 ;*
7295 ;*
7296 ;*
7297 ;*
7298 ;*
7299 ;*
7300 ;*
7301 ;*
7302 ;*
7303 ;*
7304 ;*
7305 ;*
7306 ;*
7307 ;*
7308 ;*
7309 ;*
7310 ;*
7311 ;*
7312 ;*
7313 ;*
7314 ;*
7315 ;*
7316 ;*
7317 ;*
7318 ;*
7319 ;*
7320 ;*
7321 ;*
7322 ;*
7323 ;*
7324 ;*
7325 ;*
7326 ;*
7327 ;*
7328 ;*
7329 ;*
7330 ;*
7331 ;*
7332 ;*
7333 ;*
7334 ;*
7335 ;*
7336 ;*
7337 ;*
7338 ;*
7339 ;*
7340 ;*
7341 ;*
7342 ;*
7343 ;*
7344 ;*
7345 ;*
7346 ;*
7347 ;*
7348 ;*
7349 ;*
7350 ;*
7351 ;*
7352 ;*
7353 ;*
7354 ;*
7355 ;*
7356 ;*
7357 ;*
7358 ;*
7359 ;*
7360 ;*
7361 ;*
7362 ;*
7363 ;*
7364 ;*
7365 ;*
7366 ;*
7367 ;*
7368 ;*
7369 ;*
7370 ;*
7371 ;*
7372 ;*
7373 ;*
7374 ;*
7375 ;*
7376 ;*
7377 ;*
7378 ;*
7379 ;*
7380 ;*
7381 ;*
7382 ;*
7383 ;*
7384 ;*
7385 ;*
7386 ;*
7387 ;*
7388 ;*
7389 ;*
7390 ;*
7391 ;*
7392 ;*
7393 ;*
7394 ;*
7395 ;*
7396 ;*
7397 ;*
7398 ;*
7399 ;*
7400 ;*
7401 ;*
7402 ;*
7403 ;*
7404 ;*
7405 ;*
7406 ;*
7407 ;*
7408 ;*
7409 ;*
7410 ;*
7411 ;*
7412 ;*
7413 ;*
7414 ;*
7415 ;*
7416 ;*
7417 ;*
7418 ;*
7419 ;*
7420 ;*
7421 ;*
7422 ;*
7423 ;*
7424 ;*
7425 ;*
7426 ;*
7427 ;*
7428 ;*
7429 ;*
7430 ;*
7431 ;*
7432 ;*
7433 ;*
7434 ;*
7435 ;*
7436 ;*
7437 ;*
7438 ;*
7439 ;*
7440 ;*
7441 ;*
7442 ;*
7443 ;*
7444 ;*
7445 ;*
7446 ;*
7447 ;*
7448 ;*
7449 ;*
7450 ;*
7451 ;*
7452 ;*
7453 ;*
7454 ;*
7455 ;*
7456 ;*
7457 ;*
7458 ;*
7459 ;*
7460 ;*
7461 ;*
7462 ;*
7463 ;*
7464 ;*
7465 ;*
7466 ;*
7467 ;*
7468 ;*
7469 ;*
7470 ;*
7471 ;*
7472 ;*
7473 ;*
7474 ;*
7475 ;*
7476 ;*
7477 ;*
7478 ;*
7479 ;*
7480 ;*
7481 ;*
7482 ;*
7483 ;*
7484 ;*
7485 ;*
7486 ;*
7487 ;*
7488 ;*
7489 ;*
7490 ;*
7491 ;*
7492 ;*
7493 ;*
7494 ;*
7495 ;*
7496 ;*
7497 ;*
7498 ;*
7499 ;*
7500 ;*
7501 ;*
7502 ;*
7503 ;*
7504 ;*
7505 ;*
7506 ;*
7507 ;*
7508 ;*
7509 ;*
7510 ;*
7511 ;*
7512 ;*
7513 ;*
7514 ;*
7515 ;*
7516 ;*
7517 ;*
7518 ;*
7519 ;*
7520 ;*
7521 ;*
7522 ;*
7523 ;*
7524 ;*
7525 ;*
7526 ;*
7527 ;*
7528 ;*
7529 ;*
7530 ;*
7531 ;*
7532 ;*
7533 ;*
7534 ;*
7535 ;*
7536 ;*
7537 ;*
7538 ;*
7539 ;*
7540 ;*
7541 ;*
7542 ;*
7543 ;*
7544 ;*
7545 ;*
7546 ;*
7547 ;*
7548 ;*
7549 ;*
7550 ;*
7551 ;*
7552 ;*
7553 ;*
7554 ;*
7555 ;*
7556 ;*
7557 ;*
7558 ;*
7559 ;*
7560 ;*
7561 ;*
7562 ;*
7563 ;*
7564 ;*
7565 ;*
7566 ;*
7567 ;*
7568 ;*
7569 ;*
7570 ;*
7571 ;*
7572 ;*
7573 ;*
7574 ;*
7575 ;*
7576 ;*
7577 ;*
7578 ;*
7579 ;*
7580 ;*
7581 ;*
7582 ;*
7583 ;*
7584 ;*
7585 ;*
7586 ;*
7587 ;*
7588 ;*
7589 ;*
7590 ;*
7591 ;*
7592 ;*
7593 ;*
7594 ;*
7595 ;*
7596 ;*
7597 ;*
7598 ;*
7599 ;*
7600 ;*
7601 ;*
7602 ;*
7603 ;*
7604 ;*
7605 ;*
7606 ;*
7607 ;*
7608 ;*
7609 ;*
7610 ;*
7611 ;*
7612 ;*
7613 ;*
7614 ;*
7615 ;*
7616 ;*
7617 ;*
7618 ;*
7619 ;*
7620 ;*
7621 ;*
7622 ;*
7623 ;*
7624 ;*
7625 ;*
7626 ;*
7627 ;*
7628 ;*
7629 ;*
7630 ;*
7631 ;*
7632 ;*
7633 ;*
7634 ;*
7635 ;*
7636 ;*
7637 ;*
7638 ;*
7639 ;*
7640 ;*
7641 ;*
7642 ;*
7643 ;*
7644 ;*
7645 ;*
7646 ;*
7647 ;*
7648 ;*
7649 ;*
7650 ;*
7651 ;*
7652 ;*
7653 ;*
7654 ;*
7655 ;*
7656 ;*
7657 ;*
7658 ;*
7659 ;*
7660 ;*
7661 ;*
7662 ;*
7663 ;*
7664 ;*
7665 ;*
7666 ;*
7667 ;*
7668 ;*
7669 ;*
7670 ;*
7671 ;*
7672 ;*
7673 ;*
7674 ;*
7675 ;*
7676 ;*
7677 ;*
7678 ;*
7679 ;*
7680 ;*
7681 ;*
7682 ;*
7683 ;*
7684 ;*
7685 ;*
7686 ;*
7687 ;*
7688 ;*
7689 ;*
7690 ;*
7691 ;*
7692 ;*
7693 ;*
7694 ;*
7695 ;*
7696 ;*
7697 ;*
7698 ;*
7699 ;*
7700 ;*
7701 ;*
7702 ;*
7703 ;*
7704 ;*
7705 ;*
7706 ;*
7707 ;*
7708 ;*
7709 ;*
7710 ;*
7711 ;*
7712 ;*
7713 ;*
7714 ;*
7715 ;*
7716 ;*
7717 ;*
7718 ;*
7719 ;*
7720 ;*
7721 ;*
7722 ;*
7723 ;*
7724 ;*
7725 ;*
7726 ;*
7727 ;*
7728 ;*
7729 ;*
7730 ;*
7731 ;*
7732 ;*
7733 ;*
7734 ;*
7735 ;*
7736 ;*
7737 ;*
7738 ;*
7739 ;*
7740 ;*
7741 ;*
7742 ;*
7743 ;*
7744 ;*
7745 ;*
7746 ;*
7747 ;*
7748 ;*
7749 ;*
7750 ;*
7751 ;*
7752 ;*
7753 ;*
7754 ;*
7755 ;*
7756 ;*
7757 ;*
7758 ;*
7759 ;*
7760 ;*
7761 ;*
7762 ;*
7763 ;*
7764 ;*
7765 ;*
7766 ;*
7767 ;*
7768 ;*
7769 ;*
7770 ;*
7771 ;*
7772 ;*
7773 ;*
7774 ;*
7775 ;*
7776 ;*
7777 ;*
7778 ;*
7779 ;*
7780 ;*
7781 ;*
7782 ;*
7783 ;*
7784 ;*
7785 ;*
7786 ;*
7787 ;*
7788 ;*
7789 ;*
7790 ;*
7791 ;*
7792 ;*
7793 ;*
7794 ;*
7795 ;*
7796 ;*
7797 ;*
7798 ;*
7799 ;*
7800 ;*
7801 ;*
7802 ;*
7803 ;*
7804 ;*
7805 ;*
7806 ;*
7807 ;*
7808 ;*
7809 ;*
7810 ;*
7811 ;*
7812 ;*
7813 ;*
7814 ;*
7815 ;*
7816 ;*
7817 ;*
7818 ;*
7819 ;*
7820 ;*
7821 ;*
7822 ;*
7823 ;*
7824 ;*
7825 ;*
7826 ;*
7827 ;*
7828 ;*
7829 ;*
7830 ;*
7831 ;*
7832 ;*
7833 ;*
7834 ;*
7835 ;*
7836 ;*
7837 ;*
7838 ;*
7839 ;*
7840 ;*
7841 ;*
7842 ;*
7843 ;*
7844 ;*
7845 ;*
7846 ;*
7847 ;*
7848 ;*
7849 ;*
7850 ;*
7851 ;*
7852 ;*
7853 ;*
7854 ;*
7855 ;*
7856 ;*
7857 ;*
7858 ;*
7859 ;*
7860 ;*
7861 ;*
7862 ;*
7863 ;*
7864 ;*
7865 ;*
7866 ;*
7867 ;*
7868 ;*
7869 ;*
7870 ;*
7871 ;*
7872 ;*
7873 ;*
7874 ;*
7875 ;*
7876 ;*
7877 ;*
7878 ;*
7879 ;*
7880 ;*
7881 ;*
7882 ;*
7883 ;*
7884 ;*
7885 ;*
7886 ;*
7887 ;*
7888 ;*
7889 ;*
7890 ;*
7891 ;*
7892 ;*
7893 ;*
7894 ;*
7895 ;*
7896 ;*
7897 ;*
7898 ;*
7899 ;*
7900 ;*
7901 ;*
7902 ;*
7903 ;*
7904 ;*
7905 ;*
7906 ;*
7907 ;*
7908 ;*
7909 ;*
7910 ;*
7911 ;*
7912 ;*
7913 ;*
7914 ;*
7915 ;*
7916 ;*
7917 ;*
7918 ;*
7919 ;*
7920 ;*
7921 ;*
7922 ;*
7923 ;*
7924 ;*
7925 ;*
7926 ;*
7927 ;*
7928 ;*
7929 ;*
7930 ;*
7931 ;*
7932 ;*
7933 ;*
7934 ;*
7935 ;*
7936 ;*
7937 ;*
7938 ;*
7939 ;*
7940 ;*
7941 ;*
7942 ;*
7943 ;*
7944 ;*
7945 ;*
7946 ;*
7947 ;*
7948 ;*
7949 ;*
7950 ;*
7951 ;*
7952 ;*
7953 ;*
7954 ;*
7955 ;*
7956 ;*
7957 ;*
7958 ;*
7959 ;*
7960 ;*
7961 ;*
7962 ;*
7963 ;*
7964 ;*
7965 ;*
7966 ;*
7967 ;*
7968 ;*
7969 ;*
7970 ;*
7971 ;*
7972 ;*
7973 ;*
7974 ;*
7975 ;*
7976 ;*
7977 ;*
7978 ;*
7979 ;*
7980 ;*
7981 ;*
7982 ;*
7983 ;*
7984 ;*
7985 ;*
7986 ;*
7987 ;*
7988 ;*
7989 ;*
7990 ;*
7991 ;*
7992 ;*
7993 ;*
7994 ;*
7995 ;*
7996 ;*
7997 ;*
7998 ;*
7999 ;*
8000 ;*

```

```

6923 035032 012705 177777      MOV    #MAPLNS,R5      ;PASS ACTIVE LINES BIT MASK.
6924 035036 012700 000200      MOV    #200,R0        ;PASS INERT STATE, INTERNAL LOOPBACK.
6925 035042 004767 166364      JSR     PC,WTWLNCR      ;DISABLE RECEPTION AND DMA, ETC. ON DUT.
6926 035046 012700 156430      MOV    #156430,R0      ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
6927 035052 004767 166404      JSR     PC,WTWLPR      ;WRITE INTO ALL LPR REGISTERS.
6928
6929      ;+
6930      ; SEND A NULL CHAR TO EACH LINE.
6931      ;-
6931 035056 016701 145222      MOV    IESTAT,R1      ;SET UP THE STATE OF THE INTERRUPT ENABLE BITS.
6932 035062 010177 145150      MOV    R1,BCSRA      ;SET UP THE LINE NUMBER AND INTERRUPT ENABLE
6933      ;BITS IN THE CSR.
6934 035066 112777 000000 145150      MOVB   #0,BCFDATA      ;SEND A NULL CHARACTER TO THE OUTPUT FIFO.
6935 035074 005201      INC     R1          ;NEXT LINE.
6936 035076 020127 000020      CMP    R1,#NUMLNS      ;IF ALL LINES HAVE NOT BEEN SERVICED THEN
6937 035102 002767      BLT     S#          ;BRANCH.
6938
6939      ;+
6940      ; DELAY 50 MS TO ALLOW TIME FOR THE INTERRUPT TO BE GENERATED.
6941      ;-
6941 035104 012704 000062      MOV    #50.,R4        ;PASS 50 MS TIME TO THE DELAY ROUTINE.
6942 035110 004767 162744      JSR     PC,DELAY        ;DELAY 50 MILLI-SECONDS.
6943
6944      ;+
6945      ; GENERATE A RECEPTION INTERRUPT REQUEST.
6946      ;-
6946 035114      SETVEC  RXVECA,#RXBRRT,#PRI07 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
6947      MOV    #PRI07,-(SP)
6948      MOV    #RXBRRT,-(SP)
6949      MOV    RXVECA,-(SP)
6950      MOV    #3,-(SP)
6951      TRAP   C#SVEC
6952      ADD    #10,SP
6953
6954      ;+
6955      ; SET UP FOR THE LOOP WHICH TESTS THE INTERRUPT BR LEVELS.
6956      ;-
6957 035142 012705 000340      MOV    #340,R5        ;SET UP THE PRIORITY LEVEL TO 7.
6958 035146 005003      CLR     R3          ;CLEAR THE RX PRIORITY STORE AND FLAGS.
6959 035150 005002      CLR     R2          ;CLEAR THE TX PRIORITY STORE AND FLAGS.
6960
6961      ;+
6962      ; ENABLE TX AND RX INTERRUPTS.
6963      ; PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING THE INTERRUPTS.
6964      ;-
6965 035152 004767 165030      JSR     PC,RXIE1      ;ENABLE RECEIVER INTERRUPTS.
6966 035156 004767 165606      JSR     PC,TXIE1      ;ENABLE TRANSMITTER INTERRUPTS.
6967
6968      ;+
6969      ; LOOP, LOWERING THE PROCESSOR PRIORITY UNTIL THE DUT INTERRUPTS ON RX AND TX.
6970      ;-
6971 035162 005067 145134      CLR     TXINTC        ;CLEAR THE TX INTERRUPT COUNTER.
6972 035166 005067 145132      CLR     TXINTF        ;CLEAR THE TX INTERRUPT FLAGS.
6973 035172 005067 145112      CLR     RXINTC        ;CLEAR THE RX INTERRUPT COUNTER.
6974 035176 005067 145110      CLR     RXINTF        ;CLEAR THE RX INTERRUPT FLAGS.
6975 035202      SETPRI  R5        ;SET PROCESSOR PRIORITY TO THE SELECTED VALUE.
6976      MOV    R5,R0
6977      TRAP   C#SPRI
6978
6979 035206 012704 000001      MOV    #1,R4          ;PASS 1 MS COUNT TO THE DELAY ROUTINE.
6980 035212 004767 162642      JSR     PC,DELAY        ;DELAY 1 MS TO ALLOW INTERRUPTS TO OCCUR.
6981
6982      ;+
6983      ; DETERMINE IF ANY RX DUT INTERRUPTS OCCURRED.
6984      ; LOG THE PROCESSOR PRIORITY FOR THE RX INTERRUPT IF FIRST RX INT.

```

```

6972
6973 035216 005767 145066
6974 035222 001412
6975
6976
6977
6978 035224 005703
6979 035226 001010
6980 035230 010503
6981 035232 052703 100000
6982 035236 016700 145050
6983 035242 042700 137777
6984 035246 050003
6985
6986
6987
6988
6989 035250 005767 145046
6990 035254 001405
6991
6992
6993
6994 035256 005702
6995 035260 100403
6996 035262 010502
6997 035264 052702 100000
6998
6999
7000
7001
7002 035270 162705 000040
7003 035274 002402
7004 035276 030203
7005 035300 100330
7006
7007
7008
7009 035302
035302 012700 000340
035306 104441
7010 035310
035310 016700 144712
035314 104436
7011 035316
035316 016700 144706
035322 104436
7012
7013
7014
7015
7016
7017
7018 035324 005702
7019 035326 100420
7020
7021
7022

;
; TST RXINTC ;CHECK THE RECEIVE INTERRUPT COUNTER.
; BEQ 81 ;SKIP THE PRIORITY LOG IF NO RX INT OCCURRED.
;
; IF THIS IS THE FIRST RX INTERRUPT, LOG THE PRIORITY.
;
; TST R3 ;CHECK THE RX PRIORITY STORE AND FLAGS.
; BNE 81 ;GOTO TEST FOR TX INTS IF NOT THE FIRST RX INT.
; MOV R5,R3 ;LOG THE PRESENT PRIORITY IN THE RX PRIO STORE.
; BIS #BIT15,R3 ;SET THE RX INT HAS OCCURRED FLAG.
; MOV RXINTF,R0 ;GET THE RX INTERRUPT ROUTINE FLAGS.
; BIC #137777,R0 ;CLEAR ALL BUT THE TX INT ERROR FLAG.
; BIS R0,R3 ;IF TX INT ERROR, SET BIT 14 OF THE PRIO FLAGS.
;
; DETERMINE IF ANY TX DUT INTERRUPTS HAVE OCCURRED.
; LOG THE PRESENT PROCESSOR PRIORITY IF THIS IS THE FIRST TX INTERRUPT.
;
;
81: TST TXINTC ;CHECK THE TRANSMIT INTERRUPT COUNTER.
; BEQ 101 ;SKIP THE PRIORITY LOG IF NO TX INT OCCURRED.
;
; IF THIS IS THE FIRST TX INTERRUPT, LOG THE PRIORITY.
;
; TST R2 ;CHECK THE TX PRIORITY STORE AND FLAGS.
; BMI 101 ;SKIP THE LOGGING IF NOT FIRST TX INTERRUPT.
; MOV R5,R2 ;LOG THE PRESENT PRIORITY IN THE TX PRIO STORE.
; BIS #BIT15,R2 ;SET THE TX INT HAS OCCURRED FLAG.
;
; SELECT NEXT PROCESSOR PRIORITY.
; TEST FOR BOTH RX AND TX INTERRUPTS HAVING OCCURRED, LOOP IF NOT.
;
101: SUB #40,R5 ;DECREMENT PRIORITY LEVEL BY ONE.
; BLT 121 ;GOTO CHECK FOR ERRORS IF BELOW PRIORITY ZERO.
; BIT R2,R3 ;AND PRIO FLAG TOGETHER, ALTER NONE OF THEM.
; BPL 61 ;LOOP IF RX AND TX INTS HAVEN'T BOTH OCCURRED.
;
; DISABLE INTERRUPTS AND CLEAR INTERRUPT VECTORS.
;
121: SETPRI #PRI07 ;DISABLE ALL INTERRUPTS.
;
; MOV #PRI07,R0
; TRAP C1SPRI
;
; CLRVEC RXVECA ;RETURN RX INT VECTOR TO UNUSED POOL.
; MOV RXVECA,R0
; TRAP C1CVEC
;
; CLRVEC TXVECA ;RETURN TX INT VECTOR TO UNUSED POOL.
; MOV TXVECA,R0
; TRAP C1CVEC
;
;
; VERIFY THAT RX AND TX INTERRUPTS OCCURRED,
; AT THE PROPER BR LEVEL, AND
; IN THE PROPER ORDER.
; DETERMINE IF TX INTERRUPT OCCURRED.
;
;
TST R2 ;DETERMINE WHETHER TX INT OCCURRED OR NOT.
; BMI 161 ;SKIP THESE ERRORS IF TX INT OCCURRED.
;
; DETERMINE REASON THAT NO TX INT OCCURRED.
;

```

```

7023 035330 012701 013722      MOV    #EM2610,R1      ;SELECT "NO TX INT FROM TX.ACTION" MESSAGE.
7024 035334 005777 144676      TST     @CSRA        ;CHECK THE TX.ACTION BIT OF THE DUT CSR.
7025 035340 100402              BMI     144          ;SKIP TX.ACTION CLR MSG SELECTION IF IT IS SET.
7026 035342 012701 014014      MOV     #EM2611,R1      ;SELECT "TX.ACTION CLEAR AFTER CHARS SENT" MSG.
7027                                ;REPORT "INTERRUPT BR LEVEL TEST ERROR:"
144:  ERRDF  3003,EM3001,ER0503;      >>>> ERROR #3003 <<<<.
                                TRAP    C#ERDF
                                .WORD   3003
                                .WORD   EM3001
                                .WORD   ER0503

7029
7030
7031                                ;*
7032                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7033 035356 032767 000100 144634  BIT     @BIT06.OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
7034 035364 001513              BEQ     264          ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7035
7036 035366 000427              BR      184          ;SKIP THE BR LEVEL CHECK, NO TX INT OCCURRED.
7037
7038                                ;*
7039                                ; VERIFY THAT THE TX INTERRUPT WAS AT THE PROPER BR LEVEL.
7040 035370 010204              164: MOV     R2,R4        ;CALCULATE THE BR LEVEL
7041 035372 042704 177400      BIC     #177400,R4      ; THAT THE TRANSMIT
7042 035376 006204              ASR     R4            ; INTERRUPT WAS
7043 035400 006204              ASR     R4            ; REQUESTED AT, WHICH
7044 035402 006204              ASR     R4            ; IS ONE GREATER THAN
7045 035404 006204              ASR     R4            ; THE PROCESSOR PRIORITY
7046 035406 006204              ASR     R4            ; LEVEL AT WHICH THE
7047 035410 005204              INC     R4            ; TRANSMIT INTERRUPT OCCURRED.
7048 035412 116705 144616      MOVB    BRLEVL,R5      ;GET THE EXPECTED INTERRUPT BR LEVEL.
7049 035416 120405              CMPB    R4,R5          ;COMPARE THE INTERRUPT BR LEVEL WITH EXPECTED.
7050 035420 001412              BEQ     184          ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
7051                                ;REPORT "TX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
7052 035422 012701 014326      MOV     #EM3003,R1      ;SELECT THE ERROR MESSAGE FOR THE ERROR CALL.
7053 035426              ERRDF  3004,EM3001,ER3001;      >>>> ERROR #3004 <<<<.
                                TRAP    C#ERDF
                                .WORD   3004
                                .WORD   EM3001
                                .WORD   ER3001

7054
7055                                ;*
7056                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7057 035436 032767 000100 144554  BIT     @BIT06.OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
7058 035444 001463              BEQ     264          ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7059
7060                                ;*
7061                                ; DETERMINE IF RX INTERRUPT OCCURRED.
7062                                ;*
7063 035446 005703              184: TST     R3            ;CHECK THE RX INT OCCURRED FLAG.
7064 035450 100421              BMI     224          ;SKIP THESE ERRORS IF RX INT OCCURRED.
7065
7066                                ;*
7067                                ; DETERMINE REASON THAT NO RX INT OCCURRED.
7068 035452 012701 014077      MOV     #EM2612,R1      ;SELECT "NO RX INT FROM RX.DATA.AVAIL" MSG.
7069 035456 032777 000200 144552  BIT     @BIT7,BCSRA ;CHECK THE RX.DATA.AVAIL BIT OF THE DUT CSR.
7070 035464 001002              BNE     204          ;SKIP RX.DATA.AVAIL CLR MSG IF BIT IS SET.
7071 035466 012701 014232      MOV     #EM3002,R1      ;SELECT "NO RX.DATA.AVAIL AFTER RESET" MSG.

```

```

7072      ;REPORT 'INTERRUPT BR LEVEL TEST ERROR:'
7073 035472 201:  ERRDF 3005,EM3001,ER0503;          >>>> ERROR #3005 <<<<.
      035472 104455                                TRAP  C1ERDF
      035474 005675                                .WORD 3005
      035476 014173                                .WORD EM3001
      035500 016072                                .WORD ER0503

7074      ;
7075      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7076      ;
7077 035502 032767 000100 144510      BIT  #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
7078 035510 001441      BEQ  261      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7079      ;
7080 035512 000427      BR  241      ;SKIP THE BR CHECK IF NO RX INT OCCURRED.
7081      ;
7082      ; VERIFY THAT THE RX INTERRUPT WAS AT THE PROPER BR LEVEL.
7083      ;
7084 035514 010304 221:  MOV  R3,R4      ;CALCULATE THE BR LEVEL
7085 035516 042704 177400      BIC  #177400,R4 ; THAT THE RECEIVE
7086 035522 006204      ASR  R4      ; INTERRUPT WAS
7087 035524 006204      ASR  R4      ; REQUESTED AT, WHICH
7088 035526 006204      ASR  R4      ; IS ONE GREATER THAN
7089 035530 006204      ASR  R4      ; THE PROCESSOR PRIORITY
7090 035532 006204      ASR  R4      ; LEVEL AT WHICH THE
7091 035534 005204      INC  R4      ; RECEIVE INTERRUPT OCCURRED.
7092 035536 116705 144472      MOV8 BRLEVL,R5 ;GET THE EXPECTED INTERRUPT BR LEVEL.
7093 035542 120405      CMP8 R4,R5 ;COMPARE THE INTERRUPT BR LEVEL WITH EXPECTED.
7094 035544 001412      BEQ  241      ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
7095      ;REPORT "RX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
7096 035546 012701 014402      MOV  #EM3004,R1 ;SELECT ERROR MESSAGE FOR THE ERROR CALL.
7097 035552      ERRDF 3006,EM3001,ER3001;          >>>> ERROR #3006 <<<<.
      035552 104455                                TRAP  C1ERDF
      035554 005676                                .WORD 3006
      035556 014173                                .WORD EM3001
      035560 016554                                .WORD ER3001

7098      ;
7099      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7100      ;
7101 035562 032767 000100 144430      BIT  #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
7102 035570 001411      BEQ  261      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7103      ;
7104      ;
7105      ; TEST FOR INTERRUPTS OCCURING IN THE PROPER ORDER.
7106      ;
7107 035572 032703 040000 241:  BIT  #BIT14,R3 ;CHECK THE IMPROPER INT ORDER ERROR FLAG.
7108 035576 001406      BEQ  261      ;SKIP ERROR REPORT IF ERROR DID NOT OCCUR.
7109      ;REPORT "TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS RX INT."
7110 035600 012701 014456      MOV  #EM3005,R1 ;SELECT THE ERROR MESSAGE FOR INDIRECT PRINT.
7111 035604      ERRDF 3007,EM3001,ER0503;          >>>> ERROR #3007 <<<<.
      035604 104455                                TRAP  C1ERDF
      035606 005677                                .WORD 3007
      035610 014173                                .WORD EM3001
      035612 016072                                .WORD ER0503

7112      ;
7113      ; CLEAN UP, EXIT THE TEST.
7114      ;
7115 035614 004767 165110 261:  JSR  PC,TXIE0 ;CLEAR TRANSMITTER INTERRUPTS.
7116 035620 004767 164322      JSR  PC,RXIE0 ;CLEAR RECEIVER INTERRUPTS.

```

7117 035624 005067 144452 604: CLR CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7118 035630 SETPRI #PRI07 ;DISABLE ALL INTERRUPTS.
035630 012700 000340
035634 104441
7119 035636 ENDTST
035636
035636 104401 L10054: TRAP C#ETST
7120

```

7122 .SBTTL  HARDWARE TEST          - REPBMP -
7123 ;* *****
7124 ;* - REPORT ANY BMP CODES IN THE QUEUE -
7125 ;* THIS IS A PSEUDO-TEST USED TO REPORT ANY BMP CODES THAT WERE FOUND
7126 ;* IN THE DUT'S FIFO DURING PREVIOUS TEST, AND LOGGED IN THE BMP CODE
7127 ;* QUEUE.
7128 ;* IT IS UNLIKELY THAT RUNNING THIS PSEUDO-TEST ALONE WILL PRODUCE ANY
7129 ;* ERROR REPORTS.
7130 ;*
7131 ;* *****
7132 ;* BGNTST
7133 ;*
7134 035640 000030          T24::
7135 035640 012767 000030 144452      MOV  #TNUM,TSTNUM      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
7136 035646 012767 177777 144426      MOV  #-1,CTRLCF      ;SET UP THE TEST NUMBER. (93)
7137 035654 016702 144640          MOV  BMPCQP,R2          ;INDICATE THAT WE ARE IN A TEST.
7138 035660 012703 002522          MOV  #BMPCQB,R3          ;GET THE CONTENTS OF THE POINTER.
7139 035666 001411          CMP  R2,R3          ;GET THE START ADDRESS OF THE QUEUE.
7140 ;*                               ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
7141 ;*                               ;EXIT NO CODES IN THE QUEUE.
7142 ;* THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
7143 ;*
7144 ;* ;REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN"
7145 035670 012701 015370          MOV  #EM9304,R1          ;PASS THE FIRST MESSAGE TO BE REORTED.
7146 035674 012701 015370          ERRDF 9301,EM9301,ER9301 ;>>>>> ERROR #9301 <<<<<.
7147 ;*                               TRAP  C#ERDF
7148 ;*                               .WORD 9301
7149 ;*                               .WORD EM9301
7150 ;*                               .WORD ER9301
7151 035712 005067 144364          601: CLR  CTRLCF          ;SET POINTER BACK TO THE BEGINING OF THE QUE.
7152 035716 104401          ENDTST
7153 ;*                               L10055:
7154 ;*                               TRAP  C#ETST

```

```

7154 ;*****
7155 ;
7156 ;           FVTB.HWQ
7157 ;
7158 ;*****
7160
7161
7162 .SBTTL  HARDWARE PARAMETER CODING SECTION
7163
7164
7165
7166 ;**
7167 ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
7168 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7169 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7170 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
7171 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
7172 ; WITH THE OPERATOR.
7173 ;--
7174
7175 035720      BGNHRD
7176 035720      000022
7177 035722
7178                                     .WORD  L10056-L$HARD/2
7179                                     L$HARD::
7180
7181 ;DEVICE CSR ADDRESS QUESTION:
7182 GPRMA  HWPTQ1,0,0,160000,177776,YES
7183 035722      000031
7184 035724      035766
7185 035726      160000
7186 035730      177776
7187                                     .WORD  T$CODE
7188                                     .WORD  HWPTQ1
7189                                     .WORD  T$LOLIM
7190                                     .WORD  T$HILIM
7191
7192 ;DEVICE INTERRUPT VECTOR QUESTION:
7193 GPRMA  HWPTQ2,2,0,40,776,YES
7194 035732      001031
7195 035734      036004
7196 035736      000040
7197 035740      000776
7198                                     .WORD  T$CODE
7199                                     .WORD  HWPTQ2
7200                                     .WORD  T$LOLIM
7201                                     .WORD  T$HILIM
7202
7203 ;ACTIVE LINES BIT MAP QUESTION:
7204 GPRMD  HWPTQ3,4,0,MAPLNS,0,MAPLNS,YES
7205 035742      002032
7206 035744      036037
7207 035746      177777
7208 035750      000000
7209 035752      177777
7210                                     .WORD  T$CODE
7211                                     .WORD  HWPTQ3
7212                                     .WORD  MAPLNS
7213                                     .WORD  T$LOLIM
7214                                     .WORD  T$HILIM
7215
7216 ;INTERRUPT BR LEVEL QUESTION:
7217 GPRMD  HWPTQ4,6,0,377,0,6,YES
7218 035754      003032
7219 035756      036065
7220 035760      000377
7221 035762      000C00
7222 035764      000006
7223                                     .WORD  T$CODE
7224                                     .WORD  HWPTQ4
7225                                     .WORD  377
7226                                     .WORD  T$LOLIM
7227                                     .WORD  T$HILIM
7228
7229
7230      ENDHRD
7231
7232                                     .EVEN
7233                                     L10056:
7234
7235
7236
7237
7238
7239
7240

```

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

```

7213 ;*****
7214 ;
7215 ;           FVTA.SWQ
7216 ;
7217 ;*****

7219
7220
7221 .SBTTL  SOFTWARE PARAMETER CODING SECTION
7222
7223 ;**
7224 ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
7225 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7226 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7227 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
7228 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
7229 ; WITH THE OPERATOR.
7230 ;--
7231
7232 036112          BGNSFT
7233 036112          000017
7234 036114
7235                                     .WORD  L10057-L$SOFT/2
7236                                     L$SOFT::
7237
7238 ;UNIT NUMBER PRINTOUT QUESTION:
7239 GPRML  SWPTQ1,0,20,YES
7240                                     .WORD  T$CODE
7241 036114          000130                                     .WORD  SWPTQ1
7242 036116          036152                                     .WORD  20
7243 036120          000020
7244
7245 ;ROM VERSION NUMBER PRINTOUT ON FIRST PASS QUESTION:
7246 GPRML  SWPTQ2,0,1,YES
7247                                     .WORD  T$CODE
7248 036122          000130                                     .WORD  SWPTQ2
7249 036124          036226                                     .WORD  1
7250 036126          000001
7251
7252 ;EXTENDED ERROR REPORTING QUESTION:
7253 GPRML  SWPTQ3,0,100,YES
7254                                     .WORD  T$CODE
7255 036130          000130                                     .WORD  SWPTQ3
7256 036132          036277                                     .WORD  100
7257 036134          000100
7258
7259 ; IF EXTENDED ERROR REPORTING IS NOT REQUIRED THEN SKIP THE NEXT QUESTION.
7260 ;*
7261 XFERF  ENDD
7262                                     .WORD  T$CODE
7263 036136          006044
7264
7265 ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
7266 GPRMD  SWPTQ4,2,0,177777,0,177777,YES
7267                                     .WORD  T$CODE
7268 036140          001052                                     .WORD  SWPTQ4
7269 036142          036332                                     .WORD  177777
7270 036144          177777                                     .WORD  T$LOLIM
7271 036146          000000                                     .WORD  T$HILIM
7272 036150          177777
7273
7274 .EVEN
7275
7276 ENDD:  ENDSFT
7277
7278                                     .EVEN
7279 036152          L10057:
7280
7281
7282
7283
7284
7285
7286
7287
7288
7289
7290
7291
7292
7293
7294
7295
7296
7297
7298
7299
7300

```

7266	036152	122	105	120
	036155	117	122	124
	036160	040	125	116
	036163	111	124	040
	036166	116	125	115
	036171	102	105	122
	036174	040	101	123
	036177	040	105	101
	036202	103	110	040
	036205	125	116	111
	036210	124	040	111
	036213	123	040	124
	036216	105	123	124
	036221	105	104	072
	036224	040	000	
7267	036226	122	117	115
	036231	040	126	105
	036234	122	123	111
	036237	117	116	040
	036242	120	122	111
	036245	116	124	117
	036250	125	124	040
	036253	117	116	040
	036256	124	110	105
	036261	040	106	111
	036264	122	123	124
	036267	040	120	101
	036272	123	123	072
	036275	040	000	
7268	036277	105	130	124
	036302	105	116	104
	036305	105	104	040
	036310	105	122	122
	036313	117	122	040
	036316	122	105	120
	036321	117	122	124
	036324	111	116	107
	036327	072	040	000
7269	036332	116	125	115
	036335	102	105	122
	036340	040	117	106
	036343	040	111	116
	036346	104	111	126
	036351	111	104	125
	036354	101	114	040
	036357	104	101	124
	036362	101	040	105
	036365	122	122	117
	036370	122	123	040
	036373	124	117	040
	036376	122	105	120
	036401	117	122	124
	036404	040	117	116
	036407	040	101	040
	036412	114	111	116
	036415	105	072	040
	036420	000		

SWPTQ1: .ASCIZ /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /

SWPTQ2: .ASCIZ /ROM VERSION PRINTOUT ON THE FIRST PASS: /

SWPTQ3: .ASCIZ /EXTENDED ERROR REPORTING: /

SWPTQ4: .ASCIZ /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /

7270

.EVEN

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

GPRSOB 002444 G	L\$ACP 002110 G	L10003 016070	NEWPAS 024500	SVCINS= 000001
G\$CNTD= 000200	L\$APT 002036 G	L10004 016126	NEWRES 024472	SVCSUB= 000001
G\$DELM= 000372	L\$AU 025066 G	L10005 016206	NEWSTA 024162	SVCTAG= 000001
G\$DISP= 000003	L\$AUT 002070 G	L10006 016350	NUMLNS= 000020 G	SVCTST= 000001
G\$EXCP= 000400	L\$AUTO 024736 G	L10007 016460	OOPS 020250 G	SWAPO 022356 G
G\$HILI= 000002	L\$CCP 002106 G	L10010 016552	OPTION 002220 G	SWPTQ1 036152
G\$LOLI= 000001	L\$CLEA 024740 G	L10011 016654	O\$APTS= 000000	SWPTQ2 036226
G\$NO = 000000	L\$CO 002032 G	L10012 016770	O\$AU = 000000	SWPTQ3 036277
G\$OFFS= 000400	L\$DEPO 002011 G	L10013 017060	O\$BGNR= 000001	SWPTQ4 036332
G\$OFSI= 000376	L\$DESC 004042 G	L10014 017154	O\$BGNS= 000001	S\$LSYM= 010000
G\$PRMA= 000001	L\$DESP 002076 G	L10015 017214	O\$DU = 000001	TIMER1 002340 G
G\$PRMD= 000002	L\$DEVP 002060 G	L10016 017406	O\$ERRT= 000001	TIMER2 002342 G
G\$PRML= 000000	L\$DISP 002124 G	L10017 024106	O\$GNSW= 000001	TIMER3 002344 G
G\$RADA= 000140	L\$DLY 002116 G	L10021 024734	O\$POIN= 000001	TNUM = 000030 G
G\$RAD8= 000000	L\$DTP 002040 G	L10022 024736	O\$SETU= 000000	TP4FLG 002314 G
G\$RADD= 000040	L\$DTYP 002034 G	L10023 024754	PAROA 002362 G	TP4RTN 024004 G
G\$RADL= 000120	L\$DU 024756 G	L10024 025064	PASCNT 002306 G	TP4VEC 002316 G
G\$RADO= 000020	L\$DUT 002072 G	L10025 025072	PCSL0T= 000016 G	TSABRT 022426 G
G\$XFER= 000004	L\$DVTY 004032 G	L10026 025354	PNT = 001000 G	TSTNUM 002320 G
G\$YES = 000010	L\$EF 002052 G	L10027 025604	PREGRT 004014 G	TXAD1A 002250 G
HELP = 000000	L\$ENVI 002044 G	L10030 026050	PREG05 003772	TXAD2A 002252 G
HOE = 100000 G	L\$ERRT 003762 G	L10031 026246	PRI = 002000 G	TXBFCA 002254 G
HMPTQ1 035766	L\$ETP 002102 G	L10032 026440	PRI00 = 000000 G	TXBFCA= 000016 G
HMPTQ2 036004	L\$EXP1 002046 G	L10033 026656	PRI01 = 000040 G	TXDSBL 022540 G
HMPTQ3 036037	L\$EXP4 002064 G	L10034 027064	PRI02 = 000100 G	TXENBL 022634 G
HMPTQ4 036065	L\$EXP5 002066 G	L10035 027272	PRI03 = 000140 G	TXIE0 022730 G
IBE = 010000 G	L\$HARD 035722 G	L10036 027466	PRI04 = 000200 G	TXIE1 022770 G
IDU = 000040 G	L\$HIME 002120 G	L10037 027702	PRI05 = 000240 G	TXINTC 002322 G
IER = 020000 G	L\$HPCP 002016 G	L10040 030130	PRI06 = 000300 G	TXINTF 002324 G
IESTAT 002304 G	L\$HPTP 002022 G	L10041 030424	PRI07 = 000340 G	TXINTR 024026 G
ISR = 000100 G	L\$HW 002206 G	L10042 030712	PUFIFO 020476 G	TXVECA 002230 G
IXE = 004000 G	L\$ICP 002104 G	L10043 031312	RBUFA 002240 G	T\$ARGC= 000003
I\$AU = 000041	L\$INIT 024116 G	L10044 031560	RDPDR 020560 G	T\$CODE= 001052
I\$AUTO= 000041	L\$LADP 002026 G	L10045 031750	REGTST 021032 G	T\$ERRN= 022125
I\$CLN = 000041	L\$LAST 036476 G	L10046 032266	REPSMR 021264 G	T\$EXCP= 000000
I\$DU = 000041	L\$LOAD 002100 G	L10047 032530	RESETT 021312 G	T\$FLAG= 000050
I\$HRD = 000041	L\$LUN 002074 G	L10050 032644	RMAT88 002256 G	T\$GMAN= 000000
I\$INIT= 000041	L\$MREV 002050 G	L10051 033162	ROLDAP 021424 G	T\$HILI= 177777
I\$MOD = 000041	L\$NAME 002000 G	L10052 033536	RSTRPT 021450 G	T\$LAST= 000001
I\$MSG = 000041	L\$PRIO 002042 G	L10053 034674	RXBRT 023636 G	T\$LOLI= 000000
I\$PROT= 000040	L\$PROT 024110 G	L10054 035636	RXIE0 022146 G	T\$LSYM= 010000
I\$PTAB= 000041	L\$PRT 002112 G	L10055 035716	RXIE1 022206 G	T\$LTNO= 000030
I\$PWR = 000041	L\$REPP 002062 G	L10056 035766	RXINPT 023716 G	T\$NEST= 177777
I\$RPT = 000041	L\$REV 002010 G	L10057 036152	RXINTC 002310 G	T\$NS0 = 000000
I\$SEG = 000041	L\$RPT 024102 G	MAPLNS= 177777 G	RXINTF 002312 G	T\$NS1 = 000005
I\$SETU= 000041	L\$SOFT 036114 G	MFUNIT 004070 G	RXTMA 002240 G	T\$PTNU= 000000
I\$SFT = 000041	L\$SPC 002056 G	MMENAB 002360 G	RXVECA 002226 G	T\$SAVL= 177777
I\$SRV = 000041	L\$SPCP 002020 G	MMPRES 002356 G	ROSLOT= 000002 G	T\$SEGL= 177777
I\$SUB = 000041	L\$SPTP 002024 G	MMSRO 002354 G	R1SLOT= 000004 G	T\$SUBN= 000000
I\$TST = 000041	L\$STA 002030 G	MSG1 015562 G	R2SLOT= 000006 G	T\$TAGL= 177777
J\$JMP = 000167	L\$SW 002220 G	MSG2 015640 G	R3SLOT= 000010 G	T\$TAGN= 010060
LNCTRA 002246 G	L\$TEST 002114 G	MSG3 015717 G	R4SLOT= 000012 G	T\$TEMP= 000000
LOE = 040000 G	L\$TIML 002014 G	MSLCNT 002352 G	R5SLOT= 000014 G	T\$TEST= 000030
LOT = 000010 G	L\$UNIT 002012 G	MSLGET 020120 G	SAVBMP 022232 G	T\$TSTM= 177777
LPCSLT= 000036 G	L10000 002216	MSLOOP 020234 G	SFPTBL 002220 G	T\$TSTS= 000001
LPRA = 002242 G	L10001 002224	MSTICK 002350 G	SKPSTS 022300 G	T\$AU = 010025
LPRO = 000004 G	L10002 015560	NDERPT 002222 G	SVCGBL= 000000	T\$AUT= 010022

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

. ABS. 036476 000
 000000 001

ERRORS DETECTED: 0

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