

[[[
CZDHUBO DHU 11 FUNC TST PART1 MACRO M1200 15 MAR 84 09:15 PAGE 2
PROGRAM DOCUMENT

SEQ 1

.REM 6

IDENTIFICATION

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

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 OR EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS
AFFILIATED COMPANIES.

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

DIGITAL
DEC

RDP
DEBUS

UNIBUS
DECTAPE

MASSBUS

C1
C70HUB30 PHU 11 LUNC TST PART1 MACRO M1200 15-MAR 84 09:15 PAGE 3
PROGRAM DOCUMENT

SEQ 2

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

ORIGINAL RELEASE: 15-DEC 83 ANTHONY HART

VERSION 80 3-MAR-84 ANTHONY HART

TWO NEW TESTS WERE INCLUDED IN THIS PART:

TEST 15 - CSR BIT 4 TEST.

TEST 25 - DIAGNOSTIC FIELD (BMP) TEST.

TABLE OF CONTENTS

1.0	GENERAL PROGRAM CONSIDERATIONS
1.1	PROGRAM ABSTRACT
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

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

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

1.2 SYSTEM REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DHU11 FVT:

- 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.
/PASS:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. SEE THE FLAGS SECTION OF THIS DOCUMENT.
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10 12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

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

FLAG	EFFECT
HOE	HALT ON ERROR CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IDR*	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

```
*****
START/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.
HCE	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:HOF=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

```
*****  
RESTART)/TESTS:<TEST-LIST>/PASS:<PASS CNT>/FLAGS:  
      <FLAG-LIST>/UNITS:<UNIT-LIST>  
*****
```

2.4.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE START COMMAND.

2.4.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>) - <UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE 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

```
*****  
CONTINUE)/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

PROCEED)/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

DROP)/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(NI)

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.

NL

SEQ 13

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 FAULTS 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 DHU-11. THE DEFAULT ANSWER FOR THIS QUESTION IS ADDRESS 160460 (OCTAL).
2. INTERRUPT VECTOR ADDRESS - THIS QUESTION REQUESTS THE INTERRUPT VECTOR ADDRESS OF THE SPECIFIED DHU-11. THE DEFAULT ANSWER IS 310 (OCTAL).
3. ACTIVE LINES BIT MAP - THIS QUESTION REQUESTS AN OCTAL BIT MAP OF THE SERIAL COMMUNICATION LINES ON THE DHU11 WHICH ARE BEING SELECTED FOR TESTING. IF THE BIT IN THE BIT MAP IS SET WHICH CORRESPONDS TO A PARTICULAR LINE (I.E. BIT 5 FOR LINE 5) THAT LINE WILL BE TESTED BY THE FVT.
4. BR LEVEL - THIS QUESTION REQUESTS THE INTERRUPT BR LEVEL OF THE SPECIFIED DHU-11. THE DEFAULT ANSWER IS BR 5.

2.6 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". THE SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

1. REPORT UNIT NUMBER AS EACH UNIT IS TESTED - THIS QUESTION ASKS WHETHER THE PROGRAM SHOULD REPORT THE NUMBER OF THE UNIT WHICH IT IS TESTING AS IT BEGINS TO TEST THAT UNIT.
2. ROM VERSION PRINTOUT ON THE FIRST PASS - THIS QUESTION ASKS WHETHER THE PROGRAM SHOULD PRINTOUT THE VERSIONS OF THE ON BOARD PROCESSOR ROMS DURING THE FIRST PASS OF THE PROGRAM.
3. EXTENDED ERROR REPORTING - THIS QUESTION ASKS WHETHER EXTENDED ERROR INFORMATION IS REQUIRED OTHER THAN THE "TEST FAILED" MESSAGE, ON EACH ERROR REPORTED. THE DEFAULT IS "NO" I.E. ONLY A MESSAGE REPORTING THE FACT THAT THE TEST FAILED WILL BE PRINTED.
4. NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE - THIS QUESTION IS ASKED ONLY IF THE PREVIOUS QUESTION WAS ANSWERED "YES". THE QUESTION ASKS FOR THE NUMBER OF DATA ERRORS WHICH SHOULD BE REPORTED INDIVIDUALLY BY THIS PROGRAM FOR EACH LINE FOR EACH TRANSMISSION TEST. ERRORS WHICH ARE NOT REPORTED INDIVIDUALLY ARE REPORTED IN SUMMARY ERROR REPORTS.

2.7 EXTENDED P-TABLE DIALOGUE

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

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

* UNITS (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>

E2

SEQ 17

```
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/UNIBUS AND 50HZ (IF THERE IS A CLOCK) QUESTIONS. NOTE, NOT ALL VERSIONS OF XXDP+ ASK FOR THE CLOCK FREQUENCY
3. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM
4. TYPE "START"
5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

THE GENERAL ERROR MESSAGE IS OF THE FORM:

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

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

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

THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

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

THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 SPECIFIC ERROR MESSAGES

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

CZDHU DVC FTL ERR 01603 ON UNIT 02 TST 16 SUB 000 PC: XXXXXX
DEVICE REGISTER WORD READ/WRITE TEST FAILED.

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

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

CZDHU DVC FTL ERR 01603 ON UNIT 02 TST 16 SUB 000 PC: XXXXXX
DEVICE REGISTER WORD READ/WRITE TEST FAILED.
BAD BIT(S) IN DEVICE TBUFFAD1 REGISTER FOR LINE 7 (D).
EXPECTED DATA: 000000 (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 CZDHUB:

1. DEVICE REGISTER ACCESS TEST - VERIFIES THAT THE OUT REGISTERS WILL RESPOND WITH THE CORRECT UNIBUS HANDSHAKING SIGNALS. VERIFIES THAT THE OUT 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 OUT DOES NOT IMMEDIATELY FAIL THE BACKGROUND MONITOR PROGRAM, AS THIS MAY INVALIDATE FURTHER TESTS.
10. SKIP SELFTEST TEST - VERIFIES THAT THE OUT SKIPS THE SELFTEST IN THE TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS COMPLEATION.
11. DIAGNOSTIC.FAIL (SKIP SELFTEST) TEST - VERIFIES USING THE SKIP SELFTEST SEQUENCE THAT THE DIAG.FAIL BIT GOES TO BOTH THE ACTIVE AND INACTIVE STATES WITHIN THE ALLOWED TIMES.
12. SELFTEST TEST - VERIFIES THAT THE OUT'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 OUT WILL REPORT ERRORS CORRECTLY WHEN IT IS FORCED TO FAIL.

14. ROM VERSION NUMBER - VERIFIES THAT THE ROM VERSION NUMBERS ARE REPORTED CORRECTLY AND IF REQUESTED PRINTS THEM OUT.
15. CSR BIT 4 TEST - VERIFIES THAT WHEN SET THIS BIT CAUSES THE SELFTEST TO LOOP, AND WHEN CLEARED THE SKIP SELFTEST CODES ARE RETURNED IN THE RXFIFO.
16. WORD ACCESS READ/WRITE TEST - VERIFIES THAT THE REGISTERS RESPOND CORRECTLY TO READ AND WRITE ACCESSES.
17. WORD ACCESS READ/MODIFY/WRITE TEST - VERIFIES THAT THE REGISTERS WILL RESPOND CORRECTLY TO READ/MODIFY/WRITE ACCESSES.
18. BYTE ACCESS READ/WRITE TEST - VERIFIES THAT THE REGISTERS WILL RESPOND CORRECTLY TO BYTE READ/WRITE ACCESSES.
19. BYTE ACCESS READ/MODIFY/WRITE - VERIFIES THAT THE REGISTERS WILL RESPOND CORRECTLY TO BYTE READ/MODIFY/WRITE ACCESSES.
20. ID.BIT TEST - VERIFIES THAT THE ID BIT READS AS SET.
21. TX.ENABLE (INACTIVE) TEST - VERIFIES THAT WHEN A LINE'S TX.ENBL BIT IS CLEAR, TRANSMISSION WILL NOT TAKE PLACE ON THAT LINE.
22. TX.ENABLE (ACTIVE) TEST - VERIFIES THAT WHEN A LINE'S TX.ENBL BIT IS SET, TRANSMISSION WILL TAKE PLACE ON THAT LINE.
23. INTERRUPT TEST - VERIFIES THAT THE DUT WILL GENERATE RECEPTION AND TRANSMISSION INTERRUPTS CORRECTLY.
24. BR LEVEL TEST - VERIFIES THAT THE DUT INTERRUPTS AT THE CORRECT BUS REQUEST LEVEL.
25. DIAGNOSTIC FIELD (BMP) TEST - VERIFIES THAT A REQUEST TO THE DUT TO REPORT BMP STATUS CODES IS COMPLIED WITH WITHIN THE SPECIFIED TIME. ALL ACTIVE LINES ARE TESTED.
26. REPORT BMP CODES TEST - THIS PSEUDO TEST REPORTS THE FIRST 32 BMP CHARACTERS WHICH WERE DISCOVERED IN THE FIFO DURING THE EXECUTION OF THE OTHER TESTS. THIS AVOIDS INTERRUPTION OF THE OTHER TESTS BY THESE CODES IF THEY ARE NOT CRITICAL TO THE PERFORMANCE OF THE TESTS.

6.0 EXAMPLE ERROR FREE PASS

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

.R CZDHUBO
CZDHUBO.BIN

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

CHANGE HW (L) ? Y

* UNITS (D) ? 2

UNIT 0
CSR ADDRESS: (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: (0) 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)

CZDHUBO EOP 1
0 TOTAL ERRS

DR>

```

1073          .LIST SEQ,LOC,BIN,MEB
1074          .NLST CND
1082
1083          .SBTTL  PROGRAM HEADER
1084
1085
1086
1087          .MCALL  SVC
1088 000000          SVC                      ; INITIALIZE SUPERVISOR MACROS
1089
1090          ;*****
1091          ; IF STRUCTURED MACROS ARE TO BE USED, ADD ".MCALL STRUCT" AND "STRUCT"
1092          ; TO INITIALIZE THE STRUCTURED MACROS.
1093
1094          000001          SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
1095          000001          SVCTST= 1     ; LIST TEST TAGS, SHIFTED RIGHT
1096          000001          SVCSUB= 1    ; LIST SUBTEST TAGS, SHIFTED RIGHT
1097          000001          SVCGBL= 1    ; LIST GLOBAL TAGS, SHIFTED RIGHT
1098          000001          SVCTAG= 1    ; LIST OTHER TAGS, SHIFTED RIGHT
1099
1100          ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1101          ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
1102          ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
1103          ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1104          ;*****
1105
1106 000000          .ENABL  ABS
1107          .ENABL  AMA
1108          002000          "      2000
1109
1110 002000          BGNMOD
1111
1112          ;**
1113          ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1114          ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1115          ;**
1116
1117 002000          POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136 002000          HEADER  CZDHU,B,0,16,0,PRI07
1137 002000
1138 002000          103
1139 002001          132
1140 002002          104
1141 002003          110
1142 002004          125
1143 002005          000
1144 002006          000
1145 002007          000
1146 002010

```

```

L$NAME::
          .ASCII /C
          .ASCII /Z/
          .ASCII /D/
          .ASCII /H/
          .ASCII /U/
          .BYTE  0
          .BYTE  0
          .BYTE  0
L$REV::

```


PROGRAM HEADER

```

002010      102
002011
002011      060
002012
002012 000000
002014
002014 000016
002016
002016 036700
002020
002020 037072
002022
002022 002212
002024
002024 002224
002026
002026 037454
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 004036
002062
002062 024222
002064
002064 000000
002066
002066 000000
002070
002070 000000
002072
002072 025076
002074
002074 000000
002076
002076 004046

```

```

L$DEPO::,ASCII /B/
L$UNIT::,ASCII /O/
L$TIML::,WORD 0
L$HPCP::,WORD 16
L$SPCP::,WORD L$HARD
L$HPTP::,WORD L$SOFT
L$SPTP::,WORD L$HW
L$LADP::,WORD L$SW
L$STA::,WORD L$LAST
L$CO::,WORD 0
L$DTYP::,WORD 0
L$APT::,WORD 0
L$DTP::,WORD 0
L$DTP::,WORD L$DISPATCH
L$PRIO::,WORD PRI07
L$ENVI::,WORD 0
L$EXP1::,WORD 0
L$MREV::,WORD 0
L$EF::,BYTE C$REVISION
L$EF::,BYTE C$EDIT
L$EF::,WORD 0
L$EF::,WORD 0
L$SPC::,WORD 0
L$DEVP::,WORD 0
L$REPP::,WORD L$DVTYP
L$EXP4::,WORD L$RPT
L$EXP4::,WORD 0
L$EXP5::,WORD 0
L$AUT::,WORD 0
L$DUT::,WORD 0
L$DUT::,WORD L$DC
L$UN::,WORD 0
L$DESP::,WORD L$DESC

```

N1

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 FAULTS 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.

C:\DHUBO\DHU 11 FUNC 1ST PART1 MACRO M1200 15 MAR 84 09:15 PAGE 25
DISPATCH TABLE

SEQ 27

1149
1150
1151
1152
1153
1154
1155
1156

.SBTTL DISPATCH TABLE

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

DISPATCH 26

002122
002122 000032
002124
002124 025214
002126 025476
002130 025726
002132 026172
002134 026370
002136 026562
002140 027000
002142 027206
002144 027414
002146 027610
002150 030024
002152 030252
002154 030546
002156 031034
002160 031434
002162 031776
002164 032244
002166 032434
002170 032752
002172 033214
002174 033330
002176 033646
002200 034222
002202 035360
002204 036322
002206 036616

1157

.WORD 26
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
.WORD T25
.WORD T26

DISPATCH TABLE

 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180
 1181
 1182

 1183 002210
 002210 000004
 002212
 002212

 1184
 1185 002212 160460
 1186 002214 000310
 1187 002216 177777
 1188 002220 005
 1189
 1190
 1191 002222
 002222

.SBTTL DEFAULT HARDWARE P-TABLE

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

RGNHW DFPTBL

 .WORD L10000-L\$HW/2
 L\$HW::
 DFPTBL::

 .WORD 160460 ;DEFAULT CSR ADDRESS
 .WORD 310 ;DEFAULT VECTOR ADDRESS
 .WORD 177777 ;DEFAULT ACTIVE LINES BIT MAP
 .BYTE 5 ;DEFAULT BR LEVEL
 .EVEN

ENDHW

L10000:

D3

6ZPH000 DMU-11 UNC LIST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 27

SEQ 29

1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210

1211 002222
002222 000002
002224
002224

1212
1213 002224 000001
1214 002226 000000
1215
1216 002230
002230

,SRTTL SOFTWARE P-TABLE

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

BGNSW SFPTBL

.WORD L10001-L\$SW/2

L\$SW::
SFPTBL::

OPTION:: .WORD 21
NDERPT:: .WORD 0

;BIT MAP OF PROGRAM CONTROL FLAGS
;DEFAULT NUMBER OF INDIVIDUAL DATA ERRORS TO RPT.

ENDSW

L10001:

```

1225
1226
1227
1237
1238
1239
1240
1241
1242
1243
1244
1245      000020
1246      177777
1247
1248
1249      000004
1250      000006
1251      000016
1252
1267 002230

          .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
          FLSO**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:FE17 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.  
000034 EF,PWR** 28.  
;  
; 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 LDI** 40000  
100000 HUE** 100000
```

```
; A NEW PASS HAS BEEN STARTED  
; A POWER-FAIL/POWER-UP OCCURRED
```

63

1270
 1278
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1290
 1291
 1292 002230 177777
 1293 002232 000300
 1294 002234 000304
 1295 002236 000000
 1296 002240 004
 1297
 1298
 1299
 1300
 1301
 1302 002242
 1303 002242 160020
 1304 002244 160022
 1305 002246 160024
 1306 002250 160026
 1307
 1308 002252 160030
 1309 002254 160032
 1310 002256 160034
 1311 002260 160036
 1312
 1313
 1314
 1315
 1316 002262 005464
 1317 002264 005470
 1318 002266 005475
 1319 002270 005501
 1320 002272 005517
 1321 002274 005526
 1322 002276 005537
 1323 002300 005550
 1324
 1325
 1326
 1327
 1328 002302 000000
 1329 002304 000000
 1330 002306 000000
 1331 002310 000000
 1332 002312 000000
 1333 002314 000000

.SBTTL GLOBAL DATA SECTION

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

;*****
 ; UNIT VARIABLE AREA
 ;*****

ACTLNS:: .WORD 177777 ;ACTIVE LINE BIT MAP.
 RXVECA:: .WORD 300 ;RX VECTOR ADDRESS.
 TXVECA:: .WORD 304 ;TX VECTOR ADDRESS.
 UNITN:: .WORD 0 ;UNIT NUMBER.
 BRLEVL:: .BYTE 4 ;INTERRUPT BUS REQUEST LEVEL.
 .EVEN

;*****
 ; DEVICE REGISTER ADDRESS TABLE
 ;*****

DRADRT::
 CSRA:: .WORD 160020 ;DHU-11 CSR ADDRESS.
 RXTMA:: RBUFA:: .WORD 160022 ;DHU-11 RECIEVE BUFFER/TIMER ADDRESS.
 LPRA:: .WORD 160024 ;DHU-11 LINE PARAMETER REGISTER ADDRESS.
 FDATA:: FSLSA:: .WORD 160026 ;DHU-11 FIFO SIZE/LINE STATUS REGISTER ADDRESS,
 ;AND FIFO DATA REGISTER ADDRESS.
 LNCTRA:: .WORD 160030 ;DHU-11 LINE CONTROL REGISTER ADDRESS.
 TXAD1A:: .WORD 160032 ;DHU-11 TRANSMIT BUFFER 1 REGISTER ADDRESS
 TXAD2A:: .WORD 160034 ;DHU-11 TRANSMIT BUFFER 2 REGISTER ADDRESS
 TXBFCA:: .WORD 160036 ;DHU-11 TRANSMIT BUFFER COUNT REGISTER ADDRESS

;*****
 ; REGISTER MESSAGE ADDRESS TABLE
 ;*****

RMATBB:: .WORD DRO0MG ;ADDRESS OF "CSR" MESSAGE.
 .WORD DRO2MG ;ADDRESS OF "RBUF" MESSAGE.
 .WORD DRO4MG ;ADDRESS OF "LPR" MESSAGE.
 .WORD DRO6MG ;ADDRESS OF "STAT" MESSAGE.
 .WORD DR10MG ;ADDRESS OF "LNCTRL" MESSAGE.
 .WORD DR12MG ;ADDRESS OF "TBUFAD1" MESSAGE.
 .WORD DR14MG ;ADDRESS OF "TBUFAD2" MESSAGE.
 .WORD DR16MG ;ADDRESS OF "TBUFFT" MESSAGE.

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

BUXPTR:: .WORD 0 ;STORAGE FOR RECEIVE CHARACTER BUFFER POINTER.
 EXOERR:: .WORD 0 ;"EXIT ON ERROR" FLAG.
 CTRLCF:: .WORD 0 ;STORAGE FOR THE CONTROL-C FLAG.
 IESTAT:: .WORD 0 ;STORAGE FOR THE INTERRUPT ENABLE BIT STATE.
 PASCNT:: .WORD 0 ;STORG FOR PASS COUNT USED IN ROM VERSION TEST.
 RXINTC:: .WORD 0 ;STORAGE FOR RECEIVER INTERRUPT FLAGS.


```

1334 002316 000000 RXINTF:: .WORD 0 ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
1335 002320 000000 TP4FLG:: .WORD 0 ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
1336 002322 000000 TP4VEC:: .WORD 0 ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
1337 002324 000001 TSTNUM:: .WORD 1 ;STORAGE FOR THE TEST NUMBER.
1338 002326 000000 TXINTC:: .WORD 0 ;STORAGE FOR TRANSMIT INTERRUPT COUNT.
1339 002330 000000 TXINTF:: .WORD 0 ;STORAGE FOR TRANSMIT INTERRUPT FLAGS.
1340 002332 000000 WORD1:: .WORD 0 ;LOCATION FOR PASSING INDIRECT PARAMETERS.
1341
1342
1343 ;*****
1344 ; LINE TIME CLOCK VARIABLES AND STORAGE.
1345 ;*****
1345 002334 177546 CLKCSR:: .WORD 177546 ;CSR ADDRESS OF THE LTC.
1346 002336 000300 CLKBRL:: .WORD PRIOS ;INTERRUPT PRIORITY LEVEL OF THE LTC.
1347 002340 000100 CLKVEC:: .WORD 100 ;INTERRUPT VECTOR ADDRESS OF THE LTC.
1348 002342 000074 CLKHRZ:: .WORD 60. ;INTERRUPT FREQUENCY OF THE LTC.
1349 002344 000000 TIMER1:: .WORD 0 ;HARDWARE CLOCK COUNTER #1.
1350 002346 000000 TIMER2:: .WORD 0 ;HARDWARE CLOCK COUNTER #2.
1351 002350 000170 TIMER3:: .WORD 120. ;HARDWARE BREAK COUNTER LOCATION.
1352 002352 000170 BCOUNT1:: .WORD 120. ;BREAK COUNT VALUE IN CLOCK TICKS.
1353 002354 000021 MSTICK:: .WORD 17. ;NUMBER OF MILLI-SECONDS PER LTC TICK.
1354 002356 000062 MSLCNT:: .WORD 62 ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.
1355
1356 ;*****
1357 ; MEMORY MANAGEMENT VARIABLES AND FLAGS.
1358 ;*****
1359 002360 177572 MMSRO:: .WORD 177572 ;ADDRESS OF MEM MGT STATUS REGISTER #0.
1360 002362 000000 MMPRES:: .WORD 0 ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
1361 002364 000000 MMENAB:: .WORD 0 ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).
1362 002366 172340 PAROA:: .WORD 172340 ;ADDRESS OF MEM MGT PAR #0.
1363
1364 ;*****
1365 ; BIT MASK TABLE OF UN-USED DHU DEVICE REGISTER BITS.
1366 ;*****
1367 002370 137660 UNBTB:: .WORD 137660 ;UNUSED BIT MASK FOR THE CSR
1368 002372 177777 .WORD 177777 ;UNUSED BIT MASK FOR THE RBUF/RXTIMER REG
1369 002374 000007 .WORD 7 ;UNUSED BIT MASK FOR THE LPR
1370 002376 177777 .WORD 177777 ;UNUSED BIT MASK FOR THE STAT/FIFOSIZE/DATA REG
1371 002400 166051 .WORD 166051 ;UNUSED BIT MASK FOR THE LNCTRL.
1372 002402 000000 .WORD 0 ;UNUSED BIT MASK FOR THE TBUFFA01
1373 002404 177774 .WORD 177774 ;UNUSED BIT MASK FOR THE TBUFFA02
1374 002406 000000 .WORD 0 ;UNUSED BIT MASK FOR THE TBUFFCT
1375
1376 ;*****
1377 ; TABLE OF WORDS WITH CORRESPONDING BIT SET FOR GENERATION OF BIT MAPS.
1378 ;*****
1379 002410 000001 BITTBL:: .WORD 1 ;BIT 0 SET.
1380 002412 000002 .WORD 2 ;BIT 1 SET.
1381 002414 000004 .WORD 4 ;BIT 2 SET.
1382 002416 000010 .WORD 10 ;BIT 3 SET.
1383 002420 000020 .WORD 20 ;BIT 4 SET.
1384 002422 000040 .WORD 40 ;BIT 5 SET.
1385 002424 000100 .WORD 100 ;BIT 6 SET.
1386 002426 000200 .WORD 200 ;BIT 7 SET.
1387 002430 000400 .WORD 400 ;BIT 8 SET.
1388 002432 001000 .WORD 1000 ;BIT 9 SET.
1389 002434 002000 .WORD 2000 ;BIT 10 SET.
1390 002436 004000 .WORD 4000 ;BIT 11 SET.

```

```

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

```

J3

```

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

```

GPR FRAME ACCESS EQUATES

```
1485
1486
1487
1488
1489
1490
1491
1492      000036      L.PCSLOT==      36      ;OFFSET FOR LAST RETURN PC.
1493      000016      PCSLOT==      16      ;OFFSET FOR RETURN PC.
1494      000014      R5SLOT==      14      ;OFFSET FOR R5.
1495      000012      R4SLOT==      12      ;OFFSET FOR R4.
1496      000010      R3SLOT==      10      ;OFFSET FOR R3.
1497      000006      R2SLOT==      6       ;OFFSET FOR R2.
1498      000004      R1SLOT==      4       ;OFFSET FOR R1.
1499      000002      R0SLOT==      2       ;OFFSET FOR R0.
```

.SBTTL GPR FRAME ACCESS EQUATES

;***

```
;EQUATES THAT ALLOW ACCESS TO THE STACK FRAME. THESE ARE THE
;OFFSETS INTO THE STACK FOR REGISTERS SAVED DURING THE PREG05
;ROUTINE.
```

;---

```

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

```

M3

1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573

```
.SBTTL GLOBAL MACRO DEFINITION - PASS -
*****
; THIS MACRO IS USED IN CONJUNCTION WITH THE SAVE MACRO. IT IS
; CALLED AT END OF A SUBROUTINE TO PASS PARAMETERS IN GPRS BACK TO THE
; CALLING ROUTINE BY ALTERING THE GPR SAVE AREA ON THE STACK AND THEN
; RETURNING TO PREG05 TO RESTORE THE GPRS TO THEIR SAVED VALUES.
;
; INPUTS: ONLY ALLOWED ARGUMENTS ARE "R0" THRU "R5".
; ROSLOT THRU R5SLOT MUST BE EQUATED TO THEIR RESPECTIVE GPR SAVE
; SLOT OFFSETS BEFORE CALLING THIS MACRO.
;
; OUTPUTS: THE GPR VALUES ARE PUT IN THEIR RESPECTIVE SLOTS ON THE STACK.
;
; CALLING SEQUENCE: PASS R0,R1,...
;
; COMMENTS: ANY COMBINATION OF GPR ARGUMENTS MAY BE LISTED IN ANY ORDER.
; FOR EXAMPLE, THE FOLLOWING ARE LEGAL:
; PASS R1
; PASS R4,R0,R2
; THE GPRS LISTED AS ARGUMENTS WILL BE PASSED INTACT TO THE
; CALLING ROUTINE. ALL OTHER GPRS WILL BE RESTORED.
; THE SP MUST BE AT ITS ORIGINAL VALUE WHEN PASS IS CALLED.
;
; THE MACRO CALL
; PASS R0,R3
; EXPANDS INTO THE FOLLOWING ASSEMBLY CODE:
; MOV R0,ROSLOT(SP) ;PUT R0 IN STACK SLOT.
; MOV R3,R3SLOT(SP) ;PUT R3 IN STACK SLOT.
; JSR PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
; IN THIS EXAMPLE GPRS R1, R2, R4, AND R5 WILL BE RESTORED TO
; THEIR VALUES CONTAINED IN THE STACK FRAME AND R0 AND R3
; WILL BE LEFT AT THEIR VALUES PRIOR TO THIS PASS CALL.
;
; SUBORDINATE ROUTINES CALLED: (PREGRT - LABEL WITHIN PREG05, VALUE ON STACK.)
*****

.MACRO PASS A,B,C,D,E,F
.IRP X,<A,B,C,D,E,F>
.IF NB,X
.LIST
MOV X,X'SLOT(SP) ;PUT X IN STACK SLOT.
.NLIST
.ENDC
.ENDM
.LIST
JSR PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
.NLIST
.ENDM PASS
```



```

1575 .SBTTL GLOBAL SUBROUTINE - PREG05 -
1576 ;*****
1577 ;* PRESERVE REGISTERS R0 THROUGH R5 FOR SUBROUTINE CALLS.
1578 ;*
1579 ;* INPUTS: THE RETURN ADDRESS BACK INTO THE CALLING ROUTINE MUST BE IN
1580 ;* GPR R5. (I.E. - MACROS USE "JSR R5,PREG05".)
1581 ;*
1582 ;* OUTPUTS: REGISTERS R0 THROUGH R5 ARE SAVED ON THE STACK.
1583 ;*
1584 ;* CALLING SEQUENCE: SAVE ;MACRO EXPANSION CALLS PREG05.
1585 ;* [SUBROUTINE CODE]...
1586 ;* PASS ;MACRO EXPANSION RECALLS PREG05.
1587 ;*
1588 ;* COMMENTS: THIS ROUTINE IS RE-ENTRANT.
1589 ;*
1590 ;* PARAMETERS MAY BE PASSED OUT OF A SUBROUTINE BY MODIFYING THE
1591 ;* REGISTER SAVE AREA ON THE STACK. USE THE PASS GPRN MACRO
1592 ;* TO RETURN GPR VALUES INTACT.
1593 ;* USE THE RNSLOT OFFSETS FROM THE SP TO PASS OTHER PARAMETERS.
1594 ;* [EXAMPLE: MOV VALUE,RNSLOT(SF) ]
1595 ;* MAKE SURE THE SP IS AT ITS ORIGINAL VALUE WHEN YOU DO THIS.
1596 ;*
1597 ;* SUBORDINATE ROUTINES CALLED: NONE.
1598 ;*****
1599
1600 003776 PREG05: ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
1601 003776 010446 MOV R4,-(SP) ;SAVE R4
1602 004000 010346 MOV R3,-(SP) ;SAVE R3
1603 004002 010246 MOV R2,-(SP) ;SAVE R2
1604 004004 010146 MOV R1,-(SP) ;SAVE R1
1605 004006 010046 MOV R0,-(SP) ;SAVE R0
1606 004010 010546 MOV R5,-(SP) ;PUSH RETURN PC ON TOP OF STACK
1607 004012 016605 000014 MOV R5SLOT(SP),R5 ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS
1608
1609 004016 004736 JSR PC,@(SP)+ ;CALL THE SUBROUTINE AT THE RETURN ADDRESS
1610 ;FROM THE PREG05 CALL, PUTTING THE PRESENT
1611 ;PC ON THE STACK AS A RETURN ADDRESS INTO
1612 ;THIS (PREG05) ROUTINE.
1613
1614 ;***
1615 ;THE FOLLOWING CODE IS EXECUTED WHEN THE CALLING ROUTINE DOES A
1616 ;"RETURN" [JSR PC,@(SP)+] USING THE PC DEPOSITED ON THE STACK ABOVE.
1617 ;---
1618
1619 004020 012605 PREGRT: MOV (SP)+,R5 ;PUT RETURN PC IN R5.
1620 004022 012600 MOV (SP)+,R0 ;RESTORE R0.
1621 004024 012601 MOV (SP)+,R1 ;RESTORE R1.
1622 004026 012602 MOV (SP)+,R2 ;RESTORE R2.
1623 004030 012603 MOV (SP)+,R3 ;RESTORE R3.
1624 004032 012604 MOV (SP)+,R4 ;RESTORE R4.
1625
1626 004034 000205 RTS R5 ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05.
1627 ;RESTORING R5 IN THE PROCESS.

```

CZDHUHO DHU-11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 35
GLOBAL TEXT SECTION

SEQ 40

```

1639          .SBTTL  GLOBAL TEXT SECTION
1637
1638
1639          ***
1640          ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1641          ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1642          ; MORE THAN ONE TEST.
1643          ;
1644
1645          ;
1646          ; NAMES OF DEVICES SUPPORTED BY PROGRAM
1647          ;
1648          DEVTYP  <DHU-11>
1649          004036
1650          004036      104      110      125
1651          004041      055      061      061
1652          004044      000
1653
1654          L1DEVTYPE::,ASCIIZ  /DHU-11/
1655
1656          .EVEN
1657
1658          ; TEST DESCRIPTION
1659          ;
1660          DESCRIPT  <DHU-11 FUNC TST PART1>
1661          004046
1662          004046      104      110      125
1663          004051      055      061      061
1664          004054      040      106      125
1665          004057      110      103      040
1666          004062      124      123      124
1667          004065      040      120      101
1668          004070      122      124      061
1669          004073      000
1670
1671          L1DESC::,ASCIIZ  /DHU-11 FUNC TST PAR
1672
1673          .EVEN
1674
1675          .EVEN
1676
1677

```


C4

```

1676
1677      .NLIST BIN
1678
1679      ; ***** FORMAT STATEMENTS USED IN PRINT CALLS *****
1680
1681 004074 EF0503:: .ASCIZ /#T#N/
1682 004101 EF0505:: .ASCIZ /#A      #D5#A ILLEGAL INTERRUPTS RECEIVED.#N/
1683 004154 EF1401:: .ASCIZ /#N#A ROM VERSION NUMBERS: PROC.1 = #D2#A(D)  PROC.2 = #D2#A(D)#N/
1684 004256 EF1402:: .ASCIZ /#T#A ROM VERSION NUMBER #T#N/
1685 004313 EF1601:: .ASCIZ /#A      #T#A, TEST ABORTED #N/
1686 004345 EF1602:: .ASCIZ /#A      EXPECTED DATA: #D6#A (D).#N/
1687 004407 EF1603:: .ASCIZ /#A      ACTUAL DATA:   #D6#A (D).#N/
1688 004451 EF1604:: .ASCIZ /#A      BAD BIT(S) IN DEVICE #T#A REGISTER FOR LINE #D2#A (D).#N/
1689 004546 EF3001:: .ASCIZ /#A      EXPECTED OR CORRECT VALUE: #D3#N/
1690 004615 EF3002:: .ASCIZ /#A      ACTUAL OR MEASURED  VALUE: #D3#N/
1691 004664 EF9006:: .ASCIZ /#A      #T#A #D2#A(D)#N/
1692 004710 EF9010:: .ASCIZ /#A      NUMBER OF ERRORS DETECTED ON LINE #D2#A IS #D5#N/
1693 004777 EF9016:: .ASCIZ /#A      UNEXPECTED #T#A FOR LINE #D2#A(D) IN FIFO AFTER RESET:#N/
1694 005074 EF9017:: .ASCIZ /#A      #T#A (WITH ERROR FLAGS) IS #D6#A(O)#N/
1695 005150 EF9018:: .ASCII /#A      #T#A IN SELFTEST CODE FIFO SLOT FOR LINE #D2#
1696 005230          .ASCIZ /#A(D) AFTER RESET.#N/
1697 005255 EF9301:: .ASCIZ /#A      #T#D2#A(D), BMP CODE REPORTED :#D3#A(O)#N/
1698 005333 EF9302:: .ASCIZ /#A      OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)#N/
1699 005433 MFUNIT:: .ASCIZ /#N#A TESTING UNIT :#D4#N/
1700      .EVEN
1701      .LIST BIN

```

1710
1711 .NLIST BIN

1712
1713
1714 ***** GLOBAL ERROR MESSAGES *****
1715

1716 005464 DR00MG:: .ASCIZ /CSR/
1717 005470 DR02MG:: .ASCIZ /RBUF/
1718 005475 DR04MG:: .ASCIZ /LPR/
1719 005501 DR06MG:: .ASCIZ /FIFOSIZE,STAT/
1720 005517 DR10MG:: .ASCIZ /LNCtrl/
1721 005526 DR12MG:: .ASCIZ /TBUFFAD1/
1722 005537 DR14MG:: .ASCIZ /TBUFFAD2/
1723 005550 DR16MG:: .ASCIZ /TBUFFCT/
1724 005560 EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
1725 005616 EM0201:: .ASCIZ / MASTER RESET TEST FAILED./
1726 005651 EM0202:: .ASCIZ / MASTER RESET BIT DID NOT CLEAR AFTER BOARD RESET./
1727 005735 .ASCIZ / WAITED 5 SECONDS. BIT DEFECTIVE OR FIRMWARE HUNG./
1728 006024 EM0203:: .ASCIZ / MASTER RESET BIT CLEAR IMMEDIATELY AFTER BOARD RESET./
1729 006114 .ASCIZ / BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
1730 006157 EM0204:: .ASCIZ \ MR BIT WENT CLEAR WITHIN 1/2 SECOND OF BOARD RESET.\
1731 006255 .ASCIZ / BIT DEFECTIVE OR SELFTEST WAS (INCORRECTLY) SKIPPED./
1732 006346 EM0301:: .ASCIZ /MASTER RESET (SKIP SELFTEST) TEST FAILED./
1733 006420 EM0302:: .ASCIZ / MR BIT CLR WITHIN 10 MILLISECOND AFTER BOARD RESET./
1734 006505 .ASCIZ / BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
1735 006560 EM0303:: .ASCIZ \ MR BIT WENT CLEAR 1/5 TO 5 SECONDS AFTER RESET.\
1736 006642 .ASCIZ / SELFTEST DID NOT GET SKIPPED (SHOULD HAVE BEEN SKIPPED)./
1737 006737 EM0401:: .ASCIZ /RBUF REGISTER RX CHARACTER FIELD TEST FAILED./
1738 007015 EM0402:: .ASCIZ / IMPROPER CODE FOUND IN RX FIFO AFTER DUT RESET./
1739 007077 .ASCIZ / EXPECTED: SELFTEST CODE. ACTUAL: IMPROPER CODE./
1740 007165 EM0501:: .ASCIZ /RBUF REGISTER ERROR FLAGS FIELD TEST FAILED/
1741 007241 EM0502:: .ASCIZ / RX ERROR FLAG(S) FOUND CLEAR ON SELFTEST CODE./
1742 007322 .ASCIZ / EXPECTED: ALL ERROR FLAGS SET. ACTUAL: FLAG(S) CLEAR./
1743 007415 EM0525:: .ASCIZ / RX INTERRUPT(S) RECEIVED WITH RX INTERRUPTS DISABLED./
1744 007505 EM0526:: .ASCIZ / TX INTERRUPT(S) RECEIVED WITH TX INTERRUPTS DISABLED./
1745 007575 EM0601:: .ASCIZ /CSR RX.DATA,AVAIL BIT TEST FAILED/
1746 007637 EM0602:: .ASCIZ / RX.DATA,AVAIL BIT FOUND CLEAR AFTER RESET COMPLETION./
1747 007727 .ASCIZ / EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
1748 010017 EM0603:: .ASCIZ / RX.DATA,AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO./
1749 010111 .ASCIZ / 600 CHARS READ FROM FIFO WITHOUT R.D.A BIT CLEARING./
1750 010202 EM0701:: .ASCIZ /RBUF RX.DATA.VALID BIT TEST FAILED/
1751 010245 EM0702:: .ASCIZ / RX.DATA.VALID BIT FOUND CLEAR AFTER RESET COMPLETION./
1752 010335 .ASCIZ / EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
1753 010425 EM0703:: .ASCIZ / RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO./
1754 010517 .ASCIZ / 600 CHARS READ FROM FIFO WITHOUT R.D.V BIT CLEARING./
1755 010610 EM0801:: .ASCIZ /RBUF RX.LINE.NUMBER FIELD TEST FAILED/
1756 010656 EM0802:: .ASCIZ / LINE NUMBER WRONG ON A SELFTEST CODE./
1757 010726 EM0901:: .ASCIZ /CHECK FOR BMP CODES TEST FAILED/
1758 010766 EM0902:: .ASCIZ /UNEXPECTED BMP CODES FOUND./
1759 011022 EM1001:: .ASCIZ /SKIP SELF-TEST TEST FAILED/
1760 011055 EM1002:: .ASCIZ / SKIP SELF-TEST TOOK TOO LONG TO COMPLETE, > 50 MS./
1761 011142 EM1003:: .ASCIZ / SKIP SELF-TEST COMPLETED TOO SOON, < 10 MS./
1762 011220 EM1101:: .ASCIZ /DIAGNOSTIC FAIL (SKP SELFTEST) TEST FAILED/
1763 011273 EM1201:: .ASCIZ /SELF-TEST TEST FAILED/
1764 011321 EM1202:: .ASCIZ / SELF-TEST TOOK TOO LONG TO COMPLETE, > 5 SECONDS./
1765 011405 EM1203:: .ASCIZ \ SELF-TEST COMPLETED TOO SOON, < 1/2 SECOND.\
1766 011463 EM1204:: .ASCIZ / SELF-TEST DID NOT EXECUTE/

E 4

```

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

```

GLOBAL TEXT SECTION

1832
1833
1834
1835
1836
1837
1838
1839
1840

.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.
!--

```

1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865 015510
1866 015510
1867 015510 004567 166262
1868
1869 015514 012700 000100
1870 015520 046700 164500
1871 015524 001036
1872
1873
1874
1875 015526 032705 000001
1876 015532 001410
1877 015534
1878 015534 012746 015626
1879 015540 012746 000001
1880 015544 010600
1881 015546 104414
1882 015550 062706 000004
1883 015554 032705 000002
1884 015560 001410
1885 015562
1886 015562 012746 015704
1887 015566 012746 000001
1888 015572 010600
1889 015574 104414
1890 015576 062706 000004
1891 015602
1892 015602 012746 015763
1893 015606 012746 000001
1894 015612 010600
1895 015614 104415
1896 015616 062706 000004

```

```

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

```

BGNMSG ER0101

ER0101::

SAVE

;SAVE THE GPR CONTENTS.

JSR

R5,PREG05

;CALL REGISTER SAVE SUBRT.

MOV

#BIT06,R0

;SET-UP THE BIT MAP FOR 'REPORT EXT'D ERROR INFO'

BIC

OPTION,R0

;TRY AND CLEAR THE FLAG.

RNE

6\$

;EXIT IF OPTION NOT SELECTED.

; REPORT EXTENDED ERROR INFORMATION

BIT

#BIT0,R5

;TEST FOR READ ERROR.

BEQ

2\$

;SKIP READ ERROR MSG IF NO READ ERROR.

PRINTB

#MSG1

;PRINT READ ERROR MESSAGE.

MOV

#MSG1,(SP)

MOV

#1,(SP)

MOV

SP,R0

TRAP

C\$PNTB

ADD

#4,SP

2\$:

BIT

#BIT1,R5

;TEST FOR WRITE ERROR.

BEQ

4\$

;SKIP WRITE ERROR MSG IF NO WRITE ERROR.

PRINTB

#MSG2

;PRINT WRITE ERROR MESSAGE.

MOV

#MSG2,(SP)

MOV

#1,(SP)

MOV

SP,R0

TRAP

C\$PNTB

ADD

#4,SP

4\$:

PRINTX

#MSG3

;SUGGEST THAT DHU MAY BE AT WRONG ADDRESS.

MOV

#MSG3,(SP)

MOV

#1,(SP)

MOV

SP,R0

TRAP

C\$PNTX

ADD

#4,SP

H4

GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:15 PAGE 39-1
FRO101

SEQ 46

```

1882 015622      004736      6$: PASS      JSR      ;RESTORE THE GPR CONTENTS.
1883 015624      015624      ENDMSG      PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
      015624      104423      L10002: TRAP      C$MSG
1884 015626      045      101      102 MSG1:: .ASCIZ /*ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.*N/
      015631      125      123      040
      015634      124      111      115
      015637      105      055      117
      015642      125      124      040
      015645      124      122      101
      015650      120      040      103
      015653      101      125      123
      015656      105      104      040
      015661      102      131      040
      015664      122      105      101
      015667      104      040      101
      015672      124      124      105
      015675      115      120      124
      015700      056      045      116
      015703      000
1886 015704      045      101      102 MSG2:: .ASCIZ /*ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.*N/
      015707      125      123      040
      015712      124      111      115
      015715      105      055      117
      015720      125      124      040
      015723      124      122      101
      015726      120      040      103
      015731      101      125      123
      015734      105      104      040
      015737      102      131      040
      015742      127      122      111
      015745      124      105      040
      015750      101      124      124
      015753      105      115      120
      015756      124      056      045
      015761      116      000
1887 015763      045      101      104 MSG3:: .ASCIZ /*ADHU MAY BE AT THE WRONG UNIBUS ADDRESS.*N*N/
      015766      110      125      040
      015771      115      101      131
      015774      040      102      105
      015777      040      101      124
      016002      040      124      110
      016005      105      040      127
      016010      122      117      116
      016013      107      040      125
      016016      116      111      102
      016021      125      123      040
      016024      101      104      104
      016027      122      105      123
      016032      123      056      045
      016035      116      045      116
      016040      000
1889
1889 .EVEN

```


4

CZDH030 DHU-11 FUNC 1ST PART1
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:15 PAGE 40
- ERO201 -

SEQ 47

```
1891 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ERO201
1892 *****
1893 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS 2 CONTIGUOUS
1894 ;* ASCII ERROR MESSAGES. THE ADDRESS OF THE FIRST MESSAGE IS PASSED
1895 ;* AS AN INPUT PARAMETER AND THE ADDRESS OF THE SECOND IS FOUND BY
1896 ;* SEARCHING FOR THE END OF THE FIRST MESSAGE. THE MESSAGES ARE ONLY
1897 ;* PRINTED IF EXT'D ERROR REPORTING HAS BEEN REQUESTED.
1898 ;*
1899 ;* INPUTS: R1 - ADDRESS OF THE FIRST MESSAGE TO PRINT.
1900 ;*
1901 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
1902 ;*
1903 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE FIRST MESSAGE IN R1.
1904 ;* INCLUDE THE LABEL "ERO201" AS THE MESSAGE POINTER
1905 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1906 ;*
1907 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
1908 ;* THE SECOND MESSAGE SHOULD FOLLOW THE FIRST ONE IN THE PROGRAM
1909 ;* MEMORY. EACH MESSAGE SHOULD BE DEFINED USING .ASCII?
1910 ;*
1911 ;* SUBORDINATE ROUTINES USED: NONE.
1912 *****
1913
1914 016042 BGNMSG ERO201
1915 016042 ERO201::
1916 016042 SAVE ;SAVE THE GPR CONTENTS.
1917 016042 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
1918 016046 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
1919 016052 046700 164146 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
1920 016056 001025 BNE 4, ;EXIT IF FLAG NOT SET.
1921 016060 010102 MOV R1,R2
1922 016062 105722 4$: LSTB (R2), ;CHECK FOR A ZERO BYTE (END OF MESSAGE).
1923 016064 001376 BNE 2$, ;LOOP UNTIL NEXT MESSAGE IS FOUND.
1924
1925 016066 PRINTB #EF0503,R1 ;PRINT THE FIRST MESSAGE.
1926 016066 010146 MOV R1,(SP)
1927 016070 012746 004074 MOV #EF0503,-(SP)
1928 016074 012746 000002 MOV #2,(SP)
1929 016100 010600 MOV SP,R0
1930 016102 104414 TRAP CSNTB
1931 016104 062706 000006 ADD #6,SP
1932 016110 PRINTB #EF0503,R2 ;PRINT THE SECOND MESSAGE.
1933 016110 010246 MOV R2,-(SP)
1934 016112 012746 004074 MOV #EF0503,(SP)
1935 016116 012746 000002 MOV #2,(SP)
1936 016122 010600 MOV SP,R0
1937 016124 104414 TRAP CSNTB
1938 016126 062706 000006 ADD #6,SP
1939
1940 016132 4$: PASS ;RESTORE THE GPR CONTENTS.
1941 016132 004736 JSR PC,SUSP1 ;RETURN TO PREG05 SUBRT.
1942
1943 016134 ENDMMSG
1944 016134 L10003: TRAP CSMSG
1945 016134 104423
```

GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:15 PAGE 41
- ER0503 -

SEQ 48

```

1932 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0503
1933 *****
1934 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
1935 ;* MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER, PROVIDED
1936 ;* EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
1937 ;*
1938 ;* INPUTS: R1 - ADDRESS OF THE MESSAGE TO PRINT.
1939 ;*
1940 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
1941 ;*
1942 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE MESSAGE IN R1.
1943 ;* INCLUDE THE LABEL "ER0503" AS THE MESSAGE POINTER
1944 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1945 ;*
1946 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
1947 ;*
1948 ;* SUBORDINATE ROUTINES USED: NONE.
1949 *****
1950
1951 BGNMSG ER0503
1952
1953 ER0503::
1954 MOV 0B106,R0 ;TRY TO CLEAR THE
1955 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
1956 BNE 2$ ;EXIT IF FLAG NOT SET.
1957
1958 PRINTB 0EF0503,R1 ;PRINT THE MESSAGE.
1959
1960 MOV R1,-(SP)
1961 MOV 0EF0503,-(SP)
1962 MOV 02,-(SP)
1963 MOV SP,R0
1964 TRAP C$PNTB
1965 ADD 06,SP
1966
1967 2$: ENDMMSG
1968
1969 L10004: TRAP C$MSG
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

```



```

1962 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0504 -
1963 ;*****
1964 ; THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1965 ; MESSAGES WHEN ILLEGAL INTERRUPTS ARE RECEIVED.
1966 ;
1967 ; INPUTS:      R1 - ADDRESS OF THE MESSAGE TO PRINT.
1968 ;             R2 - NUMBER OF ILLEGAL INTERRUPTS RECEIVED.
1969 ;
1970 ; OUTPUTS:     MESSAGESS ARE PRINTED AT THE OPERATOR CONSOLE.
1971 ;
1972 ; CALLING SEQUENCE:  LOAD THE ADDRESS OF THE MESSAGE IN R1.
1973 ;                   LOAD THE NUMBER OF ILLEGAL INTS IN R2.
1974 ;                   INCLUDE THE LABEL "ERR0504" AS THE MESSAGE POINTER
1975 ;                   PARAMETER IN THE DIAGNOSTIC ERROR REPORT MACRO CALL.
1976 ;
1977 ; COMMENTS:
1978 ;
1979 ; SUBORDINATE ROUTINES USED: NONE.
1980 ;*****
1981
1982 016174      BGNMSG  ER0504
1983 016174
1984 016174 012700 000100      MOV  #BIT06,R0      ;TRY TO CLEAR THE
1985 016200 046700 164020      BIC  OPTION,R0      ;EXT'D ERROR REPORTING FLAG
1986 016204 001022            BNE  2$              ;EXIT IF FLAG NOT SET.
1987
1988 016206      PRINTB  #EF0503,R1      ;PRINT THE FIRST LINE OF THE MESSAGE.
1989 016206 010146            MOV  R1,-(SP)
1990 016210 012746 004074      MOV  #EF0503,-(SP)
1991 016214 012746 000002      MOV  #2,-(SP)
1992 016220 010600            MOV  SP,R0
1993 016222 104414            TRAP  C$PNTB
1994 016224 062706 000006      ADD  #6,SP
1995
1996 016230      PRINTX  #EF0505,R2      ;PRINT THE NUMBER OF INTS RECEIVED.
1997 016230 010246            MOV  R2,-(SP)
1998 016232 012746 004101      MOV  #EF0505,-(SP)
1999 016236 012746 000002      MOV  #2,-(SP)
2000 016242 010600            MOV  SP,R0
2001 016244 104415            TRAP  C$PNTX
2002 016246 062706 000006      ADD  #6,SP
2003
2004 2$:      ENDMSG
2005
2006 016252      L10005:  TRAP  C$MSG
2007 016252 104423

```

```

1993 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER1401 -
1994 *****
1995 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1996 ;* INFORMATION (IF REQUESTED DURING THE SOFTWARE QUESTIONS) IF AN ERROR
1997 ;* IS DETECTED IN THE ROM VERSION TEST. THIS SUBROUTINE ANALYSES THE INPUT
1998 ;* PARAMETERS WHICH CONTAIN THE ROM VERSION NUMBERS FOR PROC_1 AND PROC_2
1999 ;* AND REPORTS THE APPROPRIATE ERROR MESSAGE TO THE OPERATOR.
2000 ;*
2001 ;* INPUTS: R1 - CONTAINS THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
2002 ;* R3 - CONTAINS THE ROM VERSION NUMBER OF PROC_1.
2003 ;* R4 - CONTAINS THE ROM VERSION NUMBER OF PROC_2.
2004 ;*
2005 ;* OUTPUTS: BASIC AND EXTENDED ERROR MESSAGES ARE REPORTED AT THE
2006 ;* OPERATORS CONSOLE.
2007 ;*
2008 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER1401" AS THE MESSAGE POINTER
2009 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
2010 ;*
2011 ;* COMMENTS:
2012 ;*
2013 ;* SUBORDINATE ROUTINES USED: NONE.
2014 *****
2015
2016 BGNMSG ER1401
2017 ER1401::
2018 MOV #BIT06,R0 ;TRY TO CLEAR THE
2019 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2020 BNE 60$ ;EXIT IF FLAG NOT SET.
2021
2022 PRINTB #EFC503,R1 ;REPORT THE ERROR MESSAGE PASSED IN.
2023 MOV R1,-(SP)
2024 MOV #EFC503,-(SP)
2025 MOV #2,-(SP)
2026 MOV SP,R0
2027 TRAP C$PNTB
2028 ADD #6,SP
2029
2030 ;*
2031 ;* DETERMINE WHICH ROM VERSION NUMBER(S) ARE MISSING.
2032 ;*
2033
2034 MOV #99,R5 ;GET INVALID ROM NUMBER.
2035 MOV #EM1405,R1 ;SELECT PROC_1 MESSAGE.
2036 MOV #EM1407,R2 ;SELECT THE "NOT FOUND" MESSAGE.
2037 CMPB R3,R5 ;CHECK PROC_1 ROM VERSION NUMBER.
2038 BEQ 2$ ;GO REPORT PROC_1 CODE NOT FOUND.
2039 MOV #EM1408,R2 ;SELECT "FOUND" MESSAGE.
2040 JSR PC,50$ ;GO REPORT MESSAGE.
2041
2042 MOV #EM1406,R1 ;SELECT PROC_2 MESSAGE.
2043 MOV #EM1407,R2 ;SELECT THE "NOT FOUND" MESSAGE.
2044 CMPB R4,R5 ;CHECK PROC_2 ROM VERSION NUMBER.
2045 BEQ 4$ ;GO REPORT PROC_2 CODE NOT FOUND.
2046 MOV #EM1408,R2 ;SELECT "FOUND" MESSAGE.
2047 JSR PC,50$ ;GO REPORT THE MESSAGE.
2048 BR 60$ ;EXIT.

```

CZDHUBO DHU-11 FUNC TST PART1
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:15 PAGE 43-1
ER1401 -

M4

SEQ 51

```
2043
2044 016366          50$: PRINTX 0EF1402,R1,R2 ;REPORT THE MESSAGE.
      016366 010246
      016370 010146
      016372 012746 004256
      016376 012746 000003
      016402 010600
      016404 104415
      016406 062706 000010
2045 016412 000207          RTS PC ;RETURN.
2046 016414          60$: ENDMSG
      016414
      016414 104423

MOV R2,-(SP)
MOV R1,-(SP)
MOV 0EF1402,-(SP)
MOV 03,-(SP)
MOV SP,R0
TRAP C$PRINTX
ADD 010,SP

L10006:
TRAP C$MSG
```

N4

CZDHUB0 DHU-11 FUNC 1ST PART1
GLOBAL ERROR REPORTING ROUTINEMACRO M1200 15-MAR-84 09:15 PAGE 44
- ER1601 -

SEQ 52

```

2048 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER1601
2049 ;*****
2050 ;* THIS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
2051 ;* INFORMATION IF AN ERROR IS DETECTED IN ONE OF THE DEVICE REGISTER
2052 ;* ACCESS TESTS, PROVIDED EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
2053 ;* THIS SUBROUTINE REPORTS THE ACTUAL AND EXPECTED FROM THE DEVICE
2054 ;* REGISTER(S) WHICH IS(ARE) IN FAULTY.
2055 ;*
2056 ;* INPUTS: R1 - ACTUAL DATA (UNUSED BITS SET TO 0).
2057 ;* R2 - EXPECTED DATA (UNUSED BITS SET TO 0).
2058 ;* R3 - OFFSET (IN BYTES) TO THE REGISTER BEING TESTED.
2059 ;* R5 - LINE NUMBER OF REGISTER BEING TESTED.
2060 ;* RMATBB - LABEL AT BASE OF REGISTER MESSAGE ADDRESS TABLE.
2061 ;*
2062 ;* OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
2063 ;*
2064 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER1601" AS THE MESSAGE POINTER
2065 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
2066 ;*
2067 ;* COMMENTS:
2068 ;*
2069 ;* SUBORDINATE ROUTINES CALLED: NONE
2070 ;*****
2071 016416 BGNMSG ER1601
2072 016416 ER1601::
2073 016416 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
2074 016422 046700 163576 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2075 016426 001036 BNE 2$ ;EXIT IF FLAG NOT SET.
2076
2077
2078 016430 016304 002262 MOV RMATBB(R3),R4 ;FETCH ADDRESS OF REGISTER NAME MESSAGE.
2079
2080 016434 PRINTB #EF1604,R4,R5 ;REPORT BASIC MESSAGE (REG NAME AND LINE #).
2081 016434 010546 MOV R5,-(SP)
2082 016436 010446 MOV R4,-(SP)
2083 016440 012746 004451 MOV #EF1604,-(SP)
2084 016444 012746 000003 MOV #3,(SP)
2085 016450 010600 MOV SP,R0
2086 016452 104414 TRAP C$PNTB
2087 016454 062706 000010 ADD #10,SP
2088
2089 016460 PRINTX #EF1602,R2 ;PRINT THE EXPECTED DATA.
2090 016460 010246 MOV R2,(SP)
2091 016462 012746 004345 MOV #EF1602,-(SP)
2092 016466 012746 000002 MOV #2,-(SP)
2093 016472 010600 MOV SP,R0
2094 016474 104415 TRAP C$PNTX
2095 016476 062706 000006 ADD #6,SP
2096
2097 016502 PRINTX #EF1603,R1 ;PRINT THE ACTUAL DATA.
2098 016502 010146 MOV R1,(SP)
2099 016504 012746 004407 MOV #EF1603,-(SP)
2100 016510 012746 000002 MOV #2,-(SP)
2101 016514 010600 MOV SP,R0
2102 016516 104415 TRAP C$PNTX
2103 016520 062706 000006 ADD #6,SP
2104
2105 2083 016524 2$: ENDMSG
2106 016524

```

L10007.

0
CZDHUHO DHU 11 FUNC TST PART1
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15 MAR-84 09:15 PAGE 44-1
- ER1601 -

85

SEQ 53

016524 104423

TRAP CMSG

PAR

CZDMU00 DHU 11 FUNC TEST PART1
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:15 PAGE 45
- ER1603 -

SEQ 54

2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107 016526
016526
2108 016526
016526 004567 165244
2109
2110 016532 012700 000100
2111 016536 046700 163462
2112 016542 001024
2113
2114
2115 016544
016544 010146
016546 012746 004074
016552 012746 000002
016556 010600
016560 104414
016562 062706 000006
2116
2117 016566 016702 165200
2118 016572
016572 010246
016574 012746 004313
016600 012746 000002
016604 010600
016606 104414
016610 062706 000006
2119
2120 016614
016614 004736
2121 016616
016616
016616 104423

SUBTITLE GLOBAL ERROR REPORTING ROUTINE - ER1603

THIS ERROR REPORTING ROUTINE IS USED TO PRINT OUT A BASIC ERROR
MESSAGE, ALONG WITH A MESSAGE INFORMING THE OPERATOR WHICH TEST IS
ABOUT TO BE ABORTED, PROVIDED EXTENDED ERROR INFORMATION HAS BEEN
REQUESTED. OTHERWISE ONLY A "TEST FAILURE" MESSAGE WILL BE PRINTED.

INPUTS: R1 - CONTAINS THE ADDRESS OF THE MESSAGE TO BE PRINTED.
ERRMSG - CONTAINS THE ADDRESS OF THE MESSAGE THAT INDICATES
THE TEST THAT IS BEING PERFORMED. EG DMA, BREAK ETC.

OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
"TESTNAME TEST ABORTED"

CALLING SEQUENCE: INCLUDE THE LABEL "ER1603" AS THE MESSAGE POINTER
PARAMETER IN THE DRS ERROR REPORT MACRO CALL.

COMMENTS:

SUBORDINATE ROUTINES CALLED: NONE.

BGNMSG ER1603

SAVE

JSR

ER1603::
;SAVE THE CONTENTS OF THE GPRS.
R5,PREG05 ;CALL REGISTER SAVE SUBRT.

MOV #BIT06,R0
BIC OPTION,R0
BNE 2\$

;TRY TO CLEAR THE
EXT'D ERROR REPORTING FLAG
;EXIT IF FLAG NOT SET.

PRINTB #EF0503,R1

;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.

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

MOV ERRMSG,R2
PRINTB #EF1601,R2

;GET THE "TEST MESSAGE".
;PRINT "TEST ABORTED" MESSAGE.

MOV R2,-(SP)
MOV #EF1601,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTB
ADD #6,SP

2\$ PASS

;RESTORE THE CONTENTS OF THE GPRS.
JSR PC,R(SP).

;RETURN TO PREG05 SUBRT.

ENDMSG

L10010: TRAP C#MSG

D5

```

2123 .SBIU GLOBAL ERROR REPORTING ROUTINE ER3001
2124 *****
2125 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS INTENDED FOR USE IN THE
2126 ;* INTERRUPT BR LEVEL TEST. IT REPORTS ADDITIONAL INFORMATION WHEN AN
2127 ;* INTERRUPT HAS OCCURRED AT THE WRONG BR LEVEL, UNLESS EXTENDED ERROR
2128 ;* REPORTING HAS BEEN REQUESTED, ONLY THE TEST FAIL MESSAGE
2129 ;* BE PRINTED.
2130 ;*
2131 ;* INPUTS: R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
2132 ;* R4 - BR LEVEL AT WHICH THE INT REQUEST OCCURRED.
2133 ;* R5 - EXPECTED OR CORRECT BR LEVEL FOR THE OUT.
2134 ;*
2135 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
2136 ;*
2137 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER3001" AS THE MESSAGE POINTER
2138 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2139 ;*
2140 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2141 ;*
2142 ;* SUBORDINATE ROUTINES USED: NONE.
2143 ;*
2144 ;*****
2145 BGNMSG ER3001
2146
2147 016620 016620
2148 016620 012700 000100
2149 016624 046700 163374
2150 016630 001033
2151 016632
2152 016632 010146
2153 016634 012746 004074
2154 016640 012746 000002
2155 016644 010600
2156 016646 104414
2157 016650 062706 000006
2158 016654
2159 016654 010546
2160 016656 012746 004546
2161 016662 012746 000002
2162 016666 010600
2163 016670 104415
2164 016672 062706 000006
2165 016676
2166 016676 010446
2167 016700 012746 004615
2168 016704 012746 000002
2169 016710 010600
2170 016712 104415
2171 016714 062706 000006
2172
2173 016720
2174 016720
2175 016720 104423
2176
2177 21: ENDMMSG
2178
2179 L10011: TRAP C$MSG
2180
2181 ER3001::
2182
2183 MOV 08106,R0 ;TRY TO CLEAR THE
2184 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2185 BNE 21 ;EXIT IF FLAG NOT SET.
2186
2187 PRINTB 0EF0503,R1 ;PRINT THE FIRST LINE OF THE MESSAGE.
2188 MOV R1,-(SP)
2189 MOV 0EF0503,-(SP)
2190 MOV 02,-(SP)
2191 MOV SP,R0
2192 TRAP C$PNTB
2193 ADD 06,SP
2194
2195 PRINTX 0EF3001,R5 ;REPORT EXPECTED BR LEVEL.
2196 MOV R5,(SP)
2197 MOV 0EF3001,-(SP)
2198 MOV 02,-(SP)
2199 MOV SP,R0
2200 TRAP C$PNTX
2201 ADD 06,SP
2202
2203 PRINTX 0EF3002,R4 ;REPORT ACTUAL BR LEVEL.
2204 MOV R4,-(SP)
2205 MOV 0EF3002,-(SP)
2206 MOV 02,-(SP)
2207 MOV SP,R0
2208 TRAP C$PNTX
2209 ADD 06,SP

```

E5

CZDHU30 DHU 11 FUNC 1ST PART1
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:15 PAGE 47
- ER9004 -

SEQ 56

```

2157 .BTTL GLOBAL ERROR REPORTING ROUTINE - ER9004 -
2158 *****
2159 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH REPORTS ERROR SUMMARIES
2160 ;* FOR LINES WHICH HAVE EXCEEDED THE SPECIFIED MAXIMUM NUMBER OF
2161 ;* INDIVIDUAL RECEPTION ERRORS, PROVIDED EXTENDED ERROR REPORTING HAS
2162 ;* BEEN REQUESTED BY THE OPERATOR.
2163 ;*
2164 ;* INPUTS: R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
2165 ;* ERCNTB - LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
2166 ;* ERSRPF - "REPORT ERROR SUMMARY FOR LINE" FLAGS.
2167 ;*
2168 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2169 ;*
2170 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9004" AS THE MESSAGE POINTER
2171 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2172 ;*
2173 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2174 ;* THE CONTENTS OF GPR'S R2, R3, R4, AND R5 ARE DESTROYED.
2175 ;*
2176 ;* SUBORDINATE ROUTINES USED: NONE.
2177 *****
2178
2179 BGNMSG ER9004
2180 016722 012700 000100
2181 016726 046700 163272
2182 016732 001040
2183 016734 012746 014722
2184 016734 012746 004074
2185 016744 012746 000002
2186 016750 010600
2187 016752 104414
2188 016754 062706 000006
2189 016760 005002
2190 016762 016703 163474
2191 016766 005004
2192 016770 000241
2193 016772 006003
2194 016774 103013
2195 016776 016446 002464
2196 017002 010246
2197 017004 012746 004710
2198 017010 012746 000003
2199 017014 010600
2200 017016 104415
2201 017020 062706 000010
2202 017024 012405
2203 017026 005202
2204 017030 005703
2205 017032 001356
2206 017034 017034
2207 017034 104423
2208
2209 ER9004::
2210 MOV 0BIT06,R0 ;TRY TO CLEAR THE
2211 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2212 BNE 6$ ;EXIT IF FLAG NOT SET.
2213 PRINTB 0EF0503,0EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
2214 MOV 0EM9014,-(SP)
2215 MOV 0EF0503,-(SP)
2216 MOV 02,-(SP)
2217 MOV SP,R0
2218 TRAP C$PNTB
2219 ADD 06,SP
2220
2221 CLR R2 ;CLEAR THE LINE COUNTER.
2222 MOV ERSRPF,R3 ;GET THE ERROR SUMMARY FLAGS.
2223 CLR R4 ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
2224 2$: CLC ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
2225 ROR R3 ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
2226 BCC 4$ ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
2227 PRINTX 0EF9010,R2,ERCNTB(R4)
2228 MOV ERCNTB(R4),-(SP)
2229 MOV R2,-(SP)
2230 MOV 0EF9010,-(SP)
2231 MOV 03,-(SP)
2232 MOV SP,R0
2233 TRAP C$PNTX
2234 ADD 010,SP
2235
2236 4$: MOV (R4)+,R5 ;INCREMENT THE LINE OFFSET BY 2.
2237 INC R2 ;INCREMENT THE LINE COUNTER.
2238 IST R3 ;CHECK THE ERROR SUMMARY FLAGS.
2239 BNE 2$ ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.
2240 6$: ENDMMSG
2241
2242 L10012: TRAP C$MSG

```


F5

CZDHUBO DHU-11 FUNC 1ST PART
GLOBAL ERROR REPORTING ROUTINEMACRO M1200 15-MAR-84 09:15 PAGE 48
- ER9007 -

SEQ 57

```

2197 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9007
2198 ;*****
2199 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS USED TO REPORT THAT
2200 ;* SOMETHING OTHER THAN A SELFTEST CODE WAS FOUND IN A SELFTEST CODE
2201 ;* FIFO SLOT DURING THE REMOVAL OF THE SELFTEST CODES FROM THE FIFO.
2202 ;* THIS ROUTINE IS USED BY THE RSTRPT ROUTINE. EXTENDED ERROR INFORMATION
2203 ;* IS GIVEN ONLY WHEN REQUESTED IN THE SOFTWARE QUESTIONS.
2204 ;*
2205 ;* INPUTS: R1 - ADDRESS OF ERROR MESSAGE QUALIFIER STRING.
2206 ;* R2 - INCORRECT CODE AS READ FROM THE SELFTEST CODE FIFO SLOT.
2207 ;* R3 - LINE NUMBER ASSOCIATED WITH THE SELFTEST FIFO SLOT.
2208 ;*
2209 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2210 ;*
2211 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9007" AS THE MESSAGE POINTER
2212 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2213 ;*
2214 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2215 ;*
2216 ;* SUBORDINATE ROUTINES USED: NONE.
2217 ;*****
2218
2219 017036 BGNMSG ER9007
2220 017036 ER9007::
2221 017036 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
2222 017042 046700 163156 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2223 017046 001026 BNE 2$ ;EXIT IF FLAG NOT SET.
2224
2225 017050 042703 177760 BIC #177760,R3 ;REMOVE ALL BUT LINE # BITS FROM LINE # WORD.
2226 017054 PRINTB #EF9018,R1,R3 ;REPORT SECONDARY ERROR MESSAGE.
2227 017054 010346 MOV R3,-(SP)
2228 017056 010146 MOV R1,-(SP)
2229 017060 012746 005150 MOV #EF9018,-(SP)
2230 017064 012746 000003 MOV #3,-(SP)
2231 017070 010600 MOV SP,R0
2232 017072 104414 TRAP C$PNTB
2233 017074 062706 000010 ADD #10,SP
2234
2235 017100 PRINTX #EF9017,R1,R2 ;REPORT THE ACTUAL INCORRECT CODE.
2236 017100 010246 MOV R2,-(SP)
2237 017102 010146 MOV R1,-(SP)
2238 017104 012746 005074 MOV #EF9017,-(SP)
2239 017110 012746 000003 MOV #3,-(SP)
2240 017114 010600 MOV SP,R0
2241 017116 104415 TRAP C$PNTX
2242 017120 062706 000010 ADD #10,SP
2243
2244 017124 2$: ENDMSG
2245 017124
2246 017124 104423 L10013: TRAP C$MSG

```

65

CZDHUBO DHU 11 FUNC 1ST PART1
GLOBAL ERROR REPORTING ROUTINEMACRO M1200 15-MAR-84 09:15 PAGE 49
- ER9008 -

SEQ 58

```

2231 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9008 -
2232 *****
2233 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS USED TO REPORT THAT
2234 ;* AN UNEXPECTED CODE OR CHARACTER HAS BEEN FOUND IN THE DUT RECEIVE
2235 ;* CHARACTER FIFO. THE ADDITIONAL ERROR IS REPORTED ONLY IF REQUESTED
2236 ;* DURING THE SOFTWARE QUESTIONS.
2237 ;*
2238 ;* INPUTS: R1 - ADDRESS OF PARTIAL ERROR MESSAGE STRING.
2239 ;* R2 - INCORRECT CODE AS READ FROM THE SELFTEST CODE FIFO SLOT.
2240 ;*
2241 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2242 ;*
2243 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9008" AS THE MESSAGE POINTER
2244 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2245 ;*
2246 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2247 ;*
2248 ;* SUBORDINATE ROUTINES USED: NONE.
2249 *****
2250
2251 017126 BGNMSG ER9008
2252 017126 ER9008::
2253
2253 017126 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
2254 017132 046700 163066 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2255 017136 001030 BNE 2$ ;EXIT IF FLAG NOT SET.
2256
2257 ;*
2258 ;* EXTRACT THE LINE NUMBER FROM THE INCORRECT CODE OR CHARACTER WHICH WAS READ
2259 ;* FROM THE SELFTEST CODE FIFO SLOT.
2260 ;*
2261 017140 010203 MOV R2,R3
2262 017142 000303 SWAB R3
2263 017144 042703 177760 BIC #177760,R3 ;CALCULATE LINE NUMBER OF CODE.
2264 017150 PRINTB #EF9016,R1,R3 ;REPORT TYPE OF INCORRECT CODE FOUND.
2265
2265 017150 010346 MOV R3,-(SP)
2266 017152 010146 MOV R1,-(SP)
2267 017154 012746 004777 MOV #EF9016,-(SP)
2268 017160 012746 000003 MOV #3,-(SP)
2269 017164 010600 MOV SP,R0
2270 017166 104414 TRAP C$PNTB
2271 017170 062706 000010 ADD #10,SP
2272
2272 017174 PRINTX #EF9017,R1,R2 ;REPORT THE ACTUAL INCORRECT CODE.
2273
2273 017174 010246 MOV R2,-(SP)
2274 017176 010146 MOV R1,-(SP)
2275 017200 012746 005074 MOV #EF9017,-(SP)
2276 017204 012746 000003 MOV #3,-(SP)
2277 017210 010600 MOV SP,R0
2278 017212 104415 TRAP C$PNTX
2279 017214 062706 000010 ADD #10,SP
2280
2280 017220 2$: ENDMSG
2281 017220 L10014: TRAP C$MSG
2282 017220 104423

```

H5

CZDHUBO DHU 11 FUNC TEST PART1
GLOBAL ERROR REPORTING ROUTINEMACRO M1200 15-MAR-84 09:15 PAGE 50
- ER9101 -

SEQ 59

```

2269 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9101
2270 ;*****
2271 ; THIS IS A GENERAL ERROR REPORTING SUBROUTINE WHICH REPORTS A MESSAGE
2272 ; WHICH TAKES A SINGLE, 2 DIGIT DECIMAL ARGUMENT AFTER THE END OF AN
2273 ; ASCII MESSAGE.
2274 ;*
2275 ;* INPUTS:      R1 - VALUE TO BE PRINTED AFTER MSG AS 2 DECIMAL DIGITS.
2276 ;*              R2 - ADDRESS OF MESSAGE TO PRINT FIRST.
2277 ;*
2278 ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
2279 ;*
2280 ;* CALLING SEQUENCE:  INCLUDE THE LABEL "ER9101" AS THE MESSAGE POINTER
2281 ;*                    PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2282 ;*
2283 ;* COMMENTS:      THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2284 ;*
2285 ;* SUBORDINATE ROUTINES USED: NONE.
2286 ;*****
2287
2288 017222 BGNMSG ER9101
2289 017222 ER9101::
2290 017222 012700 000100      MOV 0B106,R0      ;TRY TO CLEAR THE
2291 017226 046700 162772      BIC OPTION,R0      ;EXT'D ERROR REPORTING FLAG
2292 017232 001012      BNE 2$      ;EXIT IF FLAG NOT SET.
2293
2294
2295 017234 PRINTB 0EF9006,R2,R1 ;REPORT THE STRING FOLLOWED BY THE NUMBER.
2296 017234 010146      MOV R1,-(SP)
2297 017236 010246      MOV R2,-(SP)
2298 017240 012746 004664      MOV 0EF9006,-(SP)
2299 017244 012746 000003      MOV 03,(SP)
2300 017250 010600      MOV SP,R0
2301 017252 104414      TRAP C$PNTB
2302 017254 062706 000010      ADD 010,SP
2303
2304 2$:      ENDMMSG
2305
2306 017260 L10015:
2307 017260 104423      TRAP C$MSG

```

CZDHUBO DMU 11 FUNC 151 PART1
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:15 PAGE 51
- ER9301 -

SEQ 60

```

2299          .SBTTL GLOBAL ERROR REPORTING ROUTINE          - ER9301 -
2300          ;*****
2301          ; THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ANY BMP CODES
2302          ; THAT ARE FOUND IN THE BMP CODE QUEUE, TOGETHER WITH THE THE NUMBER OF
2303          ; THE TEST THAT WAS EXECUTING AT THE TIME THE BMP CODE WAS LOGGED.
2304          ; PROVIDED EXTENDED ERROR REPORTING HAS BEEN ENABLED.
2305          ;
2306          ; INPUTS:      R1 - THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
2307          ;              R2 - THE ADDRESS OF THE NEXT EMPTY CELL IN THE QUEUE.
2308          ;
2309          ; OUTPUTS:     THE TEST NUMBER FOLLOWED BY THE BMP CODE ARE PRINTED AT THE
2310          ;              OPERATOR CONSOLE.
2311          ;
2312          ; CALLING SEQUENCE:  INCLUDE THE LABEL "ER9301" AS THE MESSAGE POINTER
2313          ;                    PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2314          ;
2315          ; COMMENTS:     THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2316          ;
2317          ; SUBORDINATE ROUTINES USED: NONE.
2318          ;*****
2319
2320          BGNMSG ER9301
2321          SAVE                                ER9301::
2322          JSR                                ;SAVE THE GPRS ON THE STACK.
2323          ;R5,PREG05                        ;CALL REGISTER SAVE SUBRT.
2324          MOV    #BIT06,R0                  ;TRY TO CLEAR THE
2325          BIC    OPTION,R0                  ;EXT'D ERROR REPORTING FLAG
2326          BNE    #0$                         ;EXIT IF FLAG NOT SET.
2327          PRINTB #EF0503,R1                ;REPORT UNEXPECTED BMP CODES FOUND.
2328          MOV    #0$,(SP)                   ;MOV    R1,(SP)
2329          MOV    #EF0503,(SP)               ;MOV    #EF0503,(SP)
2330          MOV    #0$,(SP)                   ;MOV    #0$,(SP)
2331          MOV    SP,R0                       ;MOV    SP,R0
2332          TRAP   C$PNTB                      ;TRAP   C$PNTB
2333          ADD    #6,SP                       ;ADD    #6,SP
2334          MOV    #BMPQCB,R3                 ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
2335          MOV    #EM9302,R5                 ;GET THE MESSAGE TO BE REPORTED.
2336          2$: MOV    (R3)+,R1                ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
2337          MOV    (R3)+,R4                   ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
2338          JSR    PC,50$                     ;GO REPORT THE BMP CODE.
2339          CMP    R3,R2                      ;CHECK IF ALL CODES HAVE BEEN REPORTED.
2340          BLO    2$                         ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
2341          ;
2342          ; CHECK IF OVERFLOW HAS OCCURRED.
2343          ; THE CONDITIONS FOR OVERFLOW ARE: THE POINTER CONTAINS THE ADDRESS OF THE
2344          ; LAST CELL IN THE QUEUE, AND A BMP CODE HAS ALREADY BEEN WRITTEN INTO THAT
2345          ; CELL.
2346          ;
2347          CMP    R2,#BMPQCB-4              ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
2348          BNE    #0$                        ;EXIT IF NOT AT THE LAST LOCATION.
2349          TEST   2(R2)                      ;CHECK FOR A BMP CODE IN THE LAST CELL
2350          BEQ    #0$                        ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
2351          MOV    (R3)+,R1                   ;GET THE TEST NUMBER OFF THE QUEUE.
2352          MOV    (R3),R4                    ;GET THE BMP CODE OFF THE QUEUE.
2353          MOV    #EM9303,R5                 ;SELECT THE MESSAGE TO BE REPORTED.

```

CZDHUBO DHU-11 FUNC IST PART1
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:15 PAGE 51-1
- ER9301 -

SEQ 61

```

2348 017372                                PRINTX 0EF9302          ;REPORT OVERFLOW CONDITION.
      017372 012746 005333
      017376 012746 000001
      017402 010600
      017404 104415
      017406 062706 000004
2349 017412 004767 000002
2350 017416 000414
2351
2352 017420                                50$: PRINTX 0EF9301,R5,R1,R4 ;PRINT THE MESSAGE.
      017420 010446
      017422 010146
      017424 010546
      017426 012746 005255
      017432 012746 000004
      017436 010600
      017440 104415
      017442 062706 000012
2353 017446 000207
2354 017450                                60$: RTS      PC          ;RETURN,
      017450 004736                                PASS          ;RESTORE THE GPR CONTENTS.
2355
2356 017452                                JSR      PC,0(SP)+    ;RETURN TO PREG05 SUBRT.
      017452
      017452 104423                                L10016: TRAP  C$MSG

```

K5

CZDHU10 DHU-11 FUNC TEST PART1
GLOBAL SUBROUTINES SECTION

MACRO M1200 15-MAR-84 09:15 PAGE 52

SEQ 62

2358
2366
2367
2368
2369
2370
2371

.SBTTL GLOBAL SUBROUTINES SECTION

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

```
2373 .SBTTL GLOBAL SUBROUTINE - ALTFLD -
2374 ;* *****
2375 ;* - ALTER DEVICE REGISTER FIELDS ROUTINE -
2376 ;* THIS SUBROUTINE ALTERS THE SPECIFIED FIELD OF THE SPECIFIED DEVICE
2377 ;* REGISTER FOR THE SPECIFIED LINES. THIS ROUTINE CAN BE USED TO SET
2378 ;* OR CLEAR BITS WITHIN SELECTED FIELDS OF SELECTED REGISTERS.
2379 ;* USE EXAMPLES: SET RX.BAUD.RATE FIELDS ON LINES 3 AND 6.
2380 ;* CLEAR TX.DMA BITS ON ALL LINES.
2381 ;*
2382 ;* INPUTS: R1 - ADDRESS OF THE REGISTERS TO ALTER.
2383 ;* R2 - BIT FIELDS SET TO DESIRED STATES.
2384 ;* R3 - BIT MAP OF LINES FOR WHICH TO ALTER REGISTER.
2385 ;* R4 - MASK OF BITS TO ALTER (1 INDICATES CHANGE BIT).
2386 ;* CSRA - CONTAINS THE ADDRESS OF THE DEVICE CSR.
2387 ;* IESTAT - SAVED STATES OF THE INTERRUPT ENABLE BITS.
2388 ;*
2389 ;* OUTPUTS: DEVICE REGISTERS - SPECIFIED REGISTER FIELDS ALTERED.
2390 ;* CSR IND.ADR.REG FIELD - DESTROYED.
2391 ;*
2392 ;* CALLING SEQUENCE: JSR PC,ALTFLD
2393 ;*
2394 ;* COMMENTS: THIS ROUTINE READS THE SPECIFIED REGISTERS FOR ALL LINES
2395 ;* WITH NUMBERS LOWER THAN THE HIGHEST SPECIFIED LINE.
2396 ;* THIS ROUTINE DOES NOT READ THE CSR.
2397 ;*
2398 ;* SUBROUTINES CALLED: NONE.
2399 ;* *****
2400
2401 017454 004567 164316 ALTFLD:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2402 ; JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2403
2404 ;*
2405 ;* SET UP TO LOOP FOR EACH LINE:
2406 ;* PREPARE THE WORD TO BE ORED INTO THE REGISTER CONTENTS.
2407 ;* SET UP THE WORD TO WRITE INTO THE IND.ADR.REG FIELD OF THE CSR.
2408 017460 010400 MOV R4,R0 ;CALCULATE THE NEW CONTENTS OF THE
2409 017462 005100 COM R0 ; REGISTER FIELDS WHICH ARE TO BE
2410 017464 040002 BIC R0,R2 ; ALTERED BY THIS ROUTINE.
2411 017466 016705 162616 MOV IESTAT,R5 ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.
2412
2413 ;*
2414 ;* LOOP ONCE FOR EACH LINE, ALTERING THE SPECIFIED FIELD IN THE SPECIFIED
2415 ;* REGISTER IF THE LINE HAS BEEN SELECTED FOR ALTERING.
2416 ;* EXIT THE LOOP IF NO MORE LINES TO ALTER, OR IF WE HAVE ALTERED THE MAX
2417 ;* ALLOWABLE NUMBER OF LINES (AS SPECIFIED BY NUM.LNS).
2418 017472 000241 CLC ;PREPARE FOR ROTATE, "IST R5" DOES THIS BELOW.
2419 017474 006003 2$: ROR R5 ;GET THE LINE SELECT BIT FOR THIS LINE.
2420 017476 103006 BCC 4$ ;SKIP SETUP IF LINE IS NOT SELECTED.
2421 017500 010577 162536 MOV R5,0CSRA ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
2422 017504 011100 MOV (R1),R0 ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
2423 017506 040400 BIC R4,R0 ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
2424 017510 050200 BIS R2,R0 ;OR IN THE NEW STATES OF THE FIELDS.
2425 017512 010011 MOV R0,(R1) ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
2426 017514 005205 4$: INC R5 ;SET LINE NUMBER TO THE NEXT LINE.
2427 017516 005703 IST R3 ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
2428 017520 001365 BNE 2$ ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.
```

M5

CZDHUBO DHU 11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 53-1
GLOBAL SUBROUTINE . ALTFLD .

SEQ 64

2429
2430 017522 604: PASS JSR ;RESTORE GP'..
017522 004736 PC,@(SP);RETURN TO PREG05 SUBRT.
2431 017524 000207 RTS PC ;RETURN TO CALLING ROUTNE.


```

2433 .SBTTL GLOBAL SUBROUTINE - CALMSL -
2434 ;* *****
2435 ;* - CALIBRATE MILLI SECOND LOOP COUNT SUBROUTINE
2436 ;* THIS SUBROUTINE CALIBRATES THE TIMING LOOP WHICH IS USED IN THE MSLOOP
2437 ;* ROUTINE. THIS SUBROUTINE CALCULATES A VALUE FOR THE MSLCNT VARIABLE
2438 ;* WHICH IS THE NUMBER OF SOFTWARE LOOPS WHICH TAKES 1 MS TO EXECUTE IN
2439 ;* THE MSLOOP ROUTINE. THIS ROUTINE CALIBRATES THE COUNT BY USING THE
2440 ;* LINE TIME CLOCK (LTC), SO IF NO LTC IS AVAILABLE THE DEFAULT VALUE FOR
2441 ;* THE DELAY COUNT MUST BE USED.
2442 ;*
2443 ;*
2444 ;* INPUTS: MSLCNT - DEFAULT 1 MS DELAY LOOP COUNT VALUE, OR
2445 ;* VALUE FROM PREVIOUS CALIBRATION.
2446 ;* MSTICK - NUMBER OF MS PER LTC CLOCK TICK.
2447 ;* TIMER1 - TIMER COUNTER CHANGED BY LTC INTERRUPT SERVICE RTN.
2448 ;* CLKHRZ - NUMBER OF LTC CLICKS PER SECOND (50 OR 60).
2449 ;*
2450 ;* OUTPUTS: CARRY - SET IF LTC IS AVAILABLE, AND NEW CALIBRATION PERFORMED.
2451 ;* MSLCNT - NEW 1 MS DELAY LOOP COUNT VALUE IF LTC AVAILABLE, OR
2452 ;* UNCHANGED IF NO LTC IS AVAILABLE.
2453 ;*
2454 ;* CALLING SEQUENCE: JSR PC,CALMSL
2455 ;*
2456 ;* COMMENTS:
2457 ;*
2458 ;* SUBORDINATE ROUTINES CALLED: UNSDIV,OOPS.
2459 ;* *****
2460
2461 017526 CALMSL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2462 017526 004567 164244 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2463 017532 005067 000210 CLR 62$ ;CLEAR THE 2ND TIME FLAG.
2464 ;*
2465 ;* SYNCHRONIZE WITH THE LTC.
2466 017536 012705 000001 2$: MOV #1,R5 ;SET OUTER LOOP COUNTER TO 1 LOOP.
2467 ;* ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE ***
2468 ;* FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. ***
2469 017542 005000 CLR R0 ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
2470 017544 012767 000001 162572 MOV #1,TIMER1 ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
2471 017552 005767 162566 4$: TST TIMER1 ;CHECK FOR COUNTER HAVING GONE TO ZERO.
2472 017556 001410 BFE 6$ ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
2473 017560 005200 INC R0 ;COUNT THIS ITERATION OF THE INNER LOOP.
2474 017562 001373 BNE 4$ ;LOOP IF COUNTER HAS NOT TURNED OVER.
2475 017564 005305 DEC R5 ;DECREMENT THE INNER LOOP COUNTER.
2476 017566 003371 BGT 4$ ;LOOP IF OUTER LOOP COUNT NOT UP.
2477 ;*
2478 ;* IF WE GOT NO LTC INTERRUPT, INDICATE THAT THERE IS NO LTC AVAILABLE.
2479 ;* LTC MUST BE FLAKEY, OR NOT REALLY AN LTC AT ALL.
2480 ;*
2481 017570 005067 162546 CLR CLKHRZ ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
2482 017574 000241 CLC ;INDICATE FAILURE FOR RETURN.
2483 017576 000461 BR 60$ ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
2484 ;*
2485 ;* WE ARE NOW SYNCHRONIZED WITH THE LTC.
2486 ;* SET UP FOR THE CALIBRATION LOOP.
2487 ;*
2488 017600 012704 002344 6$: MOV #TIMER1,R4 ;WILL TEST TIMER1 IN THE LOOP BELOW.

```

```

2489 017604 005001          CLR    R1          ;CLEAR THE OUTER LOOP COUNTER.
2490 017606 005002          CLR    R2          ;INDICATE TO CHECK ALL BITS OF TIMER1.
2491 017610 005003          CLR    R3          ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2492 017612 012714 000001    MOV    #1,(R4)    ;LOAD TIMER1 WITH COUNT OF 1.
2493
2494 017616 016705 162534    81:    MOV    MSLCNT,R5    ;LOAD MS LOOP COUNT.
2495 017622 011400          101:   MOV    (R4),R0    ;GET THE TIMER1 VALUE.
2496 017624 010067 000120    MOV    R0,R4      ;SAVE WORD (LIKE IN THE REAL LOOP).
2497 017630 040200          BIC    R2,R0      ;LEAVE ALL THE BITS.
2498 017632 020003          CMP    R0,R3      ;COMPARE AGAINST ZERO.
2499 017634 000261          SEC              ;SET CARRY IN CASE OF SUCCESS.
2500 017636 001406          BEQ    121        ;EXIT LOOP IF TIMER1 HAS CLEARED.
2501 017640 005305          DEC    R5          ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2502 017642 001367          BNE    101        ;LOOP IF MS NOT UP.
2503 017644 005301          DEC    R1          ;DECREMENT THE MS TIME COUNT.
2504 017646 001353          BNE    81         ;KEEP LOOPING.
2505 017650 004767 000440    JSR     PC,00PS    ;WE OVERFLOWED, SOMETHING IS WRONG ABORT.
2506
2507
2508
2509
2510
2511
2512 017654 005401          121:   NEG    R1          ;GET NUMBER OF OUTER LOOPS.
2513 017656 016702 162474    MOV    MSLCNT,R2    ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2514 017662 010203          MOV    R2,R3      ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2515 017664 160502          SUB     R5,R2      ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2516 017666 010204          MOV    R2,R4      ;AND ADD TO ACCUMULATOR LSWORD.
2517 017670 005005          CLR    R5          ;CLEAR ACCUMULATOR MSWORD.
2518 017672 005301          141:   DEC    R1          ;CHECK R1 FOR 0 CONDITION
2519 017674 100403          RMI    161        ;SKIP MULTIPLICATION IF ZERO
2520 017676 060304          ADD     R3,R4      ;MULTIPLY NUMBER OF INNER
2521 017700 005505          ADC     R5          ;LOOPS PER OUTER LOOP BY
2522 017702 000773          BR     141        ;NUMBER OF OUTER LOOPS PERFORMED.
2523
2524
2525
2526 017704 016701 162444    161:   MOV    MSLCNT,R1    ;DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER LTC TICK.
2527 017710 010403          MOV    R4,R5      ;# OF MS PER LTC TICK IS DIVISOR.
2528 017712 010502          MOV    R5,R2      ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2529 017714 004767 003140    JSR     PC,UNSDIV    ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2530 017720 103402          BCS     181        ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK
2531 017722 004767 000366    JSR     PC,00PS    ;BYPASS OPS IF WE'RE OK.
2532 017726 010167 162424    181:   MOV    R1,MSLCNT    ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2533 017732 005167 000010    COM     #25        ;SET NEW VALUE FOR MS LOOP COUNT.
2534 017736 001277          BNE    21         ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2535 017740 000261          SEC              ;BRANCH IF ONLY ONE ITERATION DONE.
2536
2537 017742          601:   PASS          ;SET THE SUCCESS FLAG FOR EXIT.
2538 017744 000207          RTS     PC          ;RESTORE GPRS,
2539
2540 017746 000000          621:   .WORD    0          ;PC,0(0P).
2541 017750 000000          641:   .WORD    0          ;RETURN TO PREVIOUS SUBRT.

```

C:\DHUBO\DHU11\UNC1ST PART1
GLOBAL SUBROUTINE

MACRO M1200 15-MAR-84 09:15 PAGE 55
CKTRAP

Ch

SEQ 67

```

2543 .SHTTL GLOBAL SUBROUTINE - CKTRAP -
2544 *****
2545 |* CHECK TRAP ROUTINE
2546 |* THIS SUBROUTINE IS USED TO CHECK FOR A BUS TIME-OUT TRAP (004 TRAP)
2547 |* WHICH IS CAUSED BY AN ACCESS TO A NON-EXISTENT MEMORY OR I/O LOCATION.
2548 |* IF THE TRAP DOES NOT OCCUR, THIS ROUTINE RETURNS A SUCCESS INDICATION.
2549 |*
2550 |* INPUTS: R0 - SOURCE ADDRESS FOR MOVE.
2551 |* R1 - DESTINATION ADDRESS FOR MOVE.
2552 |* (R0) - SOURCE FOR THE MOVE.
2553 |*
2554 |* OUTPUTS: (R1) - WRITTEN TO THE CONTENTS OF (R0).
2555 |* CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED.
2556 |* TP4FLG - NONZERO IF TRAP OCCURRED, CLEARED OTHERWISE.
2557 |*
2558 |* CALLING SEQUENCE: JSR PC,CKTRAP
2559 |*
2560 |* COMMENTS: IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS WHICH
2561 |* IS LABELED ADRPTR WILL BE THE TRAP PC ADDRESS ON THE STACK.
2562 |*
2563 |* SUBORDINATE ROUTINES CALLED: NONE.
2564 |* *****
2565
2566 017752 CKTRAP:: SAVE R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
2567 017752 004567 164020 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2568 017756 005067 162336 CLR TP4FLG ;CLEAR THE 004 TRAP FLAGS.
2569 017762 011011 MOV (R0),(R1) ;PERFORM THE MOVE IN QUESTION.
2570 017764 005767 162330 ADRPTR:: TST TP4FLG ;CHECK FOR OCCURENCE OF TRAP.
2571 017772 001401 SEC ;INDICATE SUCCESS.
2572 017774 000241 BEQ 60$ ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
2573 017776 004736 CLC ;INDICATE FAILURE.
2574 020000 000207 60$:: PASS ;RESTORE GPRS.
2575 020000 000207 RTS PC JSR PC,(SP)+ ;RETURN TO PREG05 SUBRT.

```

D65

GLOBAL SUBROUTINE

2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619

020002
020002 004567 163770

020006 004767 001344
020012 103002

020014 004767 000522

020020
020020 004736
020022 000207

```

.SBTTL GLOBAL SUBROUTINE - CLNRST
*****
; * - CLEAN RESET OF THE DEVICE UNDER TEST
; * THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE,
; * THE DUT'S SELF-TEST IS SKIPPED, AND THE FIFO IS PURGED OF ANY ERROR
; * CODES, ETC.
; * IF THE RESET DOES NOT SUCCESSFULLY COMPLETE, THEN THE CARRY BIT IS
; * PASSED BACK TO THE CALLING ROUTINE (CLEAR).
; *
; * INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR
; * TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER,
; * ERNBR - ERROR NUMBER FOR POSSIBLE ERROR REPORT,
; * ERRBLK - ERRTP, ERNBR, AND ERRMSG SET UP CORRECTLY.
; *
; * OUTPUTS: THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
; * CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
; * ERRBLK - VALUE MAY BE DESTROYED.
; * IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
; * TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
; *
; * CALLING SEQUENCE: JSR PC,CLNRST
; *
; * COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS ERNBR.
; * THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERNBR.
; *
; * SUBORDINATE ROUTINES CALLED: DELAY,MSLGET,PUFIFO,RESETT.
*****
CLNRST:: SAVE JSR R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
;CALL REGISTER SAVE SUBRT.
; *
; * RESET THE DUT.
; * THIS ROUTINE REPORTS ERRORS WITH NUMBERS FROM ERNBR THRU ERNBR+2.
; *
; * JSR PC,RESETT ;RESET THE DUT TO A KNOWN STATE.
; * BCC 60$ ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
; *
; * PURGE THE FIFO OF ERROR CODES, SAVE ANY BMP CODES FOUND.
; *
; * JSR PC,PUFIFO ;PURGE THE FIFO.
; *
60$: ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
;RESTORE GPRS, PASS THE FOLLOWING INTACT:
;PC,2(SP), ;RETURN TO PREG05 SUBRT.
;CARRY BIT; IF CLEAR, THEN ABORT THE TEST.
; *
; * PASS JSR
; *
; * RTS PC

```

```

2621 .SBTTL GLOBAL SUBROUTINE - CLR16W -
2622 ;+ *****
2623 ;* - CLEAR SIXTEEN WORDS ROUTINE -
2624 ;* THIS SUBROUTINE CLEARS 16 WORDS STARTING WITH THE SPECIFIED WORD.
2625 ;*
2626 ;* INPUTS: RO - ADDRESS OF THE FIRST WORD TO CLEAR.
2627 ;*
2628 ;* OUTPUTS: (RO) TO (RO+15) - 16 WORDS OF MEMORY ARE CLEARED TO 0.
2629 ;*
2630 ;* CALLING SEQUENCE: JSR PC,CLR16W
2631 ;*
2632 ;* COMMENTS:
2633 ;*
2634 ;* SUBORDINATE ROUTINES CALLED: NONE.
2635 ;+ *****
2636
2637 020024 CLR16W:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2638 020024 004567 163746 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2639 020030 012701 000020 ;SET THE LOOP COUNTER TO 16.
2640 020034 005020 ;CLEAR A WORD OF MEMORY.
2641 020036 005301 ;COUNT THIS LOOP.
2642 020040 001375 ;LOOP IF NOT 16 WORD CLEARED.
2643 020042 004736 ;RESTORE GPRS.
2644 020044 000207 JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
2645 RTS PC

```

F6

CZDHUBO DHU 11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 58
GLOBAL SUBROUTINE - CNTERR -

SEQ 70

```

2645      ,SBTTL GLOBAL SUBROUTINE - CNTERR -
2646      ;* *****
2647      ;* - COUNT ERROR ROUTINE -
2648      ;* THIS SUBROUTINE IS USED TO COUNT A "DATA" ERROR ON THE SPECIFIED
2649      ;* LINE. IT CHECKS WHETHER ERROR SUMMARY REPORTING IS ACTIVE, OR SHOULD
2650      ;* BE MADE ACTIVE ON THIS LINE, AND ACTIVATES IT IF NECESSARY.
2651      ;*
2652      ;* INPUTS:      R5 - LINE NUMBER OF LINE UNDER CONSIDERATION.
2653      ;*              ERCNTB - LABEL AT BASE OF ERROR COUNTERS TABLE.
2654      ;*              ERSMRP - ERROR SUMMARY FLAGS (BIT SET IF LINE IN SUMMARY MODE).
2655      ;*              NDERPT - NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE.
2656      ;*
2657      ;* OUTPUTS:     CARRY - SET IF LINE IS IN ERROR SUMMARY MODE.
2658      ;*              ERCNT - ERROR COUNTER INCREMENTED FOR SPECIFIED LINE.
2659      ;*              ERSMRP - BIT SET IF LINE SHOULD BE IN SUMMARY MODE.
2660      ;*
2661      ;* CALLING SEQUENCE: JSR PC,CNTERR
2662      ;*
2663      ;* COMMENTS:
2664      ;*
2665      ;* SUBORDINATE ROUTINES CALLED: NONE.
2666      ;* *****
2667
2668 020046 CNTERR:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
2669 020046 004567 163724 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2670
2671 ;* COUNT THE ERROR ON THE COUNTER FOR THE SPECIFIED LINE.
2672 ;*
2672 020052 006305 ASL R5 ;FORM WORD OFFSET FROM LINE NUMBER.
2673 020054 016501 002464 MOV ERCNTB(R5),R1 ;GET THE PRESENT ERROR COUNT FOR THIS LINE.
2674 020060 005201 INC R1 ;COUNT ERROR.
2675 020062 103402 BCS 2$ ;OVERFLOW? YES, DON'T UPDATE COUNTER IN TABLE.
2676 020064 010165 002464 MOV R1,ERCNTB(R5) ;UPDATE ERROR COUNTER TABLE ENTRY.
2677 020070 005767 162132 2$: TST NDERPT
2678 020074 001411 BEQ 60$ ;SUMMARYS DISABLED? YES, EXIT WITH CARRY 0.
2679 020076 020167 162124 CMP R1,NDERPT ;NO, CHECK FOR ENOUGH ERRORS FOR SUMMARY USE.
2680 020102 101002 BHI 4$ ;ENOUGH ERRORS TO USE SUMMARY? YES, GO HANDLE.
2681 020104 000241 CLC ;INDICATE NOT TO USE SUMMARY REPORT YET.
2682 020106 000404 BR 60$ ;EXIT WITH CARRY 0.
2683 020110 056567 002410 162344 4$: BIS BITTBL(R5),ERSMRP ;SET THE ERROR SUMMARY FLAG FOR LINE.
2684 020116 000261 SEC ;INDICATE TO USE SUMMARY REPORT.
2685 020120 60$: PASS ;RESTORE GPRS.
2686 020122 000207 RTS PC JSR PC,8(SP) ;RETURN TO PREG05 SUBRT.

```



```

2688 .SBTTL GLOBAL SUBROUTINE - DELAY -
2689 ;*****
2690 ;* - DELAY SUBROUTINE -
2691 ;* THIS SUBROUTINE IS USED TO DELAY A VARIABLE NUMBER OF MILLI-SECONDS.
2692 ;*
2693 ;* INPUTS: R4 - CONTAINS THE NUMBER OF MS TO DELAY.
2694 ;* MSLCNT.
2695 ;*
2696 ;* OUTPUTS: NONE.
2697 ;*
2698 ;* CALLING SEQUENCE: JSR PC,DELAY
2699 ;*
2700 ;* COMMENTS: IF NO HARDWARE CLOCK INTERRUPTS ARE OCCURRING, CONTROL-CS WILL
2701 ;* NOT BE HONORED FOR THE DURATION OF THE DELAY.
2702 ;*
2703 ;* SUBORDINATE ROUTINES CALLED: NONE.
2704 ;*****
2705
2706 020124 DELAY:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
020124 004567 163646 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2707 020130 010401 MOV R4,R1 ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
2708 020132 012702 MOV #1,R2 ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
2709 020136 005003 CLR R3 ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
2710 020140 012704 020162 MOV #62,R4 ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
2711 020144 004767 000130 JSR PC,MSLOOP ;DELAY THE REQUESTED # OF MS.
2712 020150 103002 BCC 60$ ;EXIT ROUTINE IF WE TIMED-OUT.]
2713 020152 004767 000136 JSR PC,OOPS ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
2714 020156 60$: PASS ;RESTORE GPRS.
020156 004736 JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
2715 020160 000207 RTS PC
2716
2717 020162 177777 62$: .WORD -1 ;DUMMY, NON-ZERO WORD.

```

H6

SEQ 72

```

2719 .SBTTL GLOBAL SUBROUTINE - MSLGET -
2720 *****
2721 ; - MILLI SECONDS LOOP WHICH RETURNS READ WORD AND REMAINING TIME -
2722 ; THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
2723 ; TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD. THE
2724 ; CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
2725 ; DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
2726 ; THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
2727 ; ROUTINE AND THEN ONCE EACH MILLI-SECOND THERE AFTER.
2728 ; UPON RETURN, THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION
2729 ; IS RETURNED BY THIS SUBROUTINE.
2730 ;
2731 ; INPUTS: R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
2732 ; R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
2733 ; R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
2734 ; R4 - ADDRESS OF THE WORD TO TEST.
2735 ; MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
2736 ;
2737 ; OUTPUTS: R0 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
2738 ; R1 - REMAINING NUMBER OF MS IN TIME-OUT TIME.
2739 ; CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
2740 ;
2741 ; CALLING SEQUENCE: JSR PC,MSLGET
2742 ;
2743 ; COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
2744 ; CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
2745 ; ON THE SYSTEM.
2746 ; THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
2747 ; DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
2748 ; LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
2749 ; IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
2750 ; THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
2751 ; IF THE CONDITION IS MET, FAILURE OTHERWISE.
2752 ;
2753 ;
2754 ; SUBORDINATE ROUTINES CALLED: NONE.
2755 *****
2756
2757 020164 MSLGET:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2758 020164 004567 163606 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2759 ;
2760 ; SET UP MASK FOR REMOVING UNUSED BITS IN THE TEST WORD, AND CLEAR UNUSED
2761 ; BITS IN THE DESIRED STATE WORD TO ALLOW DIRECT COMPARISON.
2762 020170 005102 COM R2 ;GET MASK OF UNUSED BITS.
2763 020172 040203 BIC R2,R3 ;MASK OUT UNUSED BITS IN DESIRED STATE WORD.
2764 ;
2765 ; HANDLE THE TEST AND EXIT IF WE HAVE A 0 TIME-OUT VALUE.
2766 ;
2767 020174 005701 TST R1 ;TEST THE TIME-OUT VALUE FOR ZERO.
2768 020176 001011 BNE 27 ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
2769 020200 011400 MOV (R4),R0 ;GET THE WORD TO TEST BEFORE EXITING.
2770 020202 010067 000070 MOV R0,R2 ;SAVE VALUE SO WE CAN RETURN IT.
2771 020206 040200 BIC R2,R0 ;MASK OUT UNTESTED BITS OF WORD.
2772 020210 020007 CMP R0,R3 ;COMPARE AGAINST DESIRED STATE WORD.
2773 020212 000261 SEC ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
2774 020214 001420 BEQ 67 ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.

```



```

2775 020216 000241          CLC          ;INDICATE FAILURE (TIME-OUT).
2776 020220 000416          BR          6$ ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
2777                          ;*
2778                          ; NON-ZERO TIME-OUT VALUE. LOOP, WAITING FOR CONDITION OR TIME-OUT.
2779                          ;*
2780 020222 016705 162130    2$: MOV      MSLCNT,R5 ;LOAD MS LOOP COUNT.
2781 020226 011400          4$: MOV      (R4),R0 ;GET THE WORD TO TEST.
2782 020230 010067 000042    MOV      R0,62$ ;SAVE WORD IN CASE THIS IS THE LAST.
2783 020234 040200          BIC      R2,R0 ;MASK OUT UNTESTED BITS OF WORD.
2784 020236 020003          CMP      R0,R3 ;COMPARE AGAINST DESIRED STATE WORD.
2785 020240 000261          SEC          ;SET CARRY IN CASE OF SUCCESS.
2786 020242 001405          BEQ      6$ ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
2787 020244 005305          DEC      R5 ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2788 020246 001367          BNE      4$ ;LOOP IF MS NOT UP.
2789 020250 005301          DEC      R1 ;DECREMENT THE MS TIME COUNT.
2790 020252 001363          BNE      2$ ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
2791 020254 000241          CLC          ;CLEAR CARRY, WE TIMED-OUT.
2792                          ;*
2793                          ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME OUT VALUE).
2794                          ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD. EXIT ROUTINE.
2795                          ;*
2796 020256 016700 000014    6$: MOV      62$,R0 ;PASS OUT THE LAST READ WORD.
2797 020262          60$: PASS      R0,R1 ;RESTORE GPRS, EXCEPT THE FOLLOWING:
                                MOV      R0,R0SLOT(SP) ;PUT R0 IN STACK SLOT.
                                MOV      R1,R1SLOT(SP) ;PUT R1 IN STACK SLOT.
                                JSR      PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
2798                          ;R0 - LAST READ WORD CHECKED FOR CONDITION.
2799                          ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
2800 020274 000207          RTS      PC ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
2801                          ;*
2802                          ; LOCAL STORAGE.
2803                          ;
2804 020276 000000          62$: .WORD 0 ;STORAGE FOR THE LAST READ WORD.

```

J6

CZDHUB0 DHU-11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 61
 GLOBAL SUBROUTINE - MSLOOP -

SEQ 74

```

2806 .SBTTL GLOBAL SUBROUTINE - MSLOOP -
2807 ;*****
2808 ;* - TEST LOOP SUBROUTINE -
2809 ;* THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
2810 ;* TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD. THE
2811 ;* CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
2812 ;* DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
2813 ;* THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
2814 ;* ROUTINE AND THEN ONCE EACH MILLI-SECOND THEREAFTER.
2815 ;*
2816 ;* INPUTS: R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
2817 ;* R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
2818 ;* R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
2819 ;* R4 - ADDRESS OF THE WORD TO TEST.
2820 ;* MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
2821 ;*
2822 ;* OUTPUTS: CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
2823 ;*
2824 ;* CALLING SEQUENCE: JSR PC,MSLOOP
2825 ;*
2826 ;* COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
2827 ;* CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
2828 ;* ON THE SYSTEM.
2829 ;* THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
2830 ;* DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
2831 ;* LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
2832 ;* IF A TIME OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
2833 ;* THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
2834 ;* IF THE CONDITION IS MET, FAILURE OTHERWISE.
2835 ;*
2836 ;* SUBORDINATE ROUTINES CALLED: MSLGET.
2837 ;*****
2838
2839 020300 MSLOOP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5
2840 020300 004567 163472 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2841
2842 ;*
2843 ;* CALLING THE MSLGET ROUTINE FROM THE MSLOOP ROUTINE ISOLATES THE CALLER OF
2844 ;* MSLOOP FROM THE RETURNED TEST WORD AND REMAINING TIME-OUT VALUES.
2845 020304 004767 177654 JSR PC,MSLGET ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.
2846
2847 020310 004736 60$: PASS ;RESTORE GPRS,
2848 020312 000207 RTS PC JSR PC,REGSP1 ;RETURN TO PREG05 SUBRT.
;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.

```

K6

CZDHUB0 DHU-11 FUNC 1ST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 62
GLOBAL SUBROUTINE - OOPS -

SEQ 75

```

2850 .SBTIL GLOBAL SUBROUTINE - OOPS -
2851 ;* *****
2852 ;* PROGRAM ABORT SUBROUTINE
2853 ;* THIS SUBROUTINE IS USED TO ABORT THE PROGRAM WHEN A FATAL ERROR IS
2854 ;* DETECTED IN THE PROGRAM OR THE HOST SYSTEM HARDWARE. AN ERROR MESSAGE
2855 ;* IS PRINTED GIVING SOME INFORMATION ABOUT THE NATURE OF THE ABORT.
2856 ;*
2857 ;* INPUTS: R1 - ERROR CODE GIVING REASON FOR ABORT.
2858 ;*
2859 ;* OUTPUTS: AN ERROR MESSAGE IS PRINTED.
2860 ;* A LIST OF RETURN PC VALUES FOR ALL SUBROUTINE CALLS IS PRINTED.
2861 ;*
2862 ;* CALLING SEQUENCE: JSR PC,OOPS
2863 ;*
2864 ;* COMMENTS:
2865 ;*
2866 ;* SUBORDINATE ROUTINES CALLED: NONE.
2867 ;* *****
2868
2869 020314 OOPS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2870 020314 004567 163456 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2871 ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
2872 ERRSF 101,EM0101
2873 020320 TRAP C$ERSF
2874 020320 104454 .WORD 101
2875 020322 000145 .WORD EM0101
2876 020324 020360 .WORD 0
2877 020326 000000
2878 ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
2879 020330 PRINTF 0EM0102
2880 020330 012746 020444 MOV 0EM0102,-(SP)
2881 020334 012746 000001 MOV 01,-(SP)
2882 020340 010600 MOV SP,R0
2883 020342 104417 TRAP C$PNIF
2884 020344 062706 000004 ADD 04,SP
2885 2$: BREAK ;LOOK FOR OPERATOR CONTROL-C INPUT.
2886 020350 TRAP C$BRK
2887 020352 104422
2888 020354 000776 ;INFINITE LOOP.
2889 020354 004736 ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
2890 020356 000207 JSR PC,0(SP); ;RETURN TO PREG05 SUBRT.
2891 ; ROUTINE IN THE FUTURE, SO BE CONSISTANT,
2892
2893 EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
2894 020360 110 117 123
2895 020363 124 040 103
2896 020366 117 115 120
2897 020371 125 124 105
2898 020374 122 040 110
2899 020377 101 122 104
2900 020402 127 101 122
2901 020405 105 040 117
2902 020410 122 040 123
2903 020413 117 106 124
2904 020416 127 101 122
2905 020421 105 040 102
2906 020424 125 107 040
2907 020427 105 116 103
2908 020432 117 125 116
2909 020435 124 105 122

```

CZDHUBO DHU 11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 62-1
GLOBAL SUBROUTINE - OOPS -

SEQ 76

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

.EVEN

```

2883 .SBTTL GLOBAL SUBROUTINE - PUFIFO -
2884 *****
2885 ;* - PURGE THE FIFO
2886 ;* THIS ROUTINE TRIES TO REMOVE ALL THE CHARACTERS FROM THE FIFO,
2887 ;* ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE BMP CODE QUEUE.
2888 ;*
2889 ;* INPUTS: RBUFA- CONTAINS THE ADDRESS OF THE RECEIVER.
2890 ;*
2891 ;*
2892 ;* OUTPUTS: CARRY BIT - INDICATES THE STATE OF THE FIFO, SET:= PURGED.
2893 ;* BMPCQ - THE CONTENTS OF THE BMP CODE QUEUE MAY BE UPDATED.
2894 ;*
2895 ;* CALLING SEQUENCE: JSR PC,PUFIFO
2896 ;*
2897 ;* COMMENTS:
2898 ;*
2899 ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
2900 *****
2901
2902 020542 PUFIFO::SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2903 020543 004567 163230 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2904 020544 012701 001000 ;SET MAXIMUM TRY COUNT OF 512.
2905 020545 016704 161466 ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
2906 020546 011402 ;
2907 020547 100016 2$: MOV (R4),R2 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
2908 ; BPL 6$ ;EXIT IF THE FIFO IS EMPTY, DATA VALID CLR.
2909 ;
2910 ; CHECK IF THE READ CHARACTER IS ACTUALLY A BMP CODE.
2911 ; IF IT IS, THEN SAVE IT ON THE BMP CODE QUEUE TO BE REPORTED LATER.
2912 020548 012700 070000 ;
2913 020549 040200 ; MOV #70000,R0 ;GENERATE A BIT MAP OF CHAR ERROR BITS
2914 020550 001006 ; BIC R2,R0 ; WHICH ARE NOT SET FOR CHAR.
2915 ; BNE 4$ ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
2916 ;
2917 ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
2918 020551 012700 000301 ;
2919 020552 040200 ; MOV #301,R0 ; CHECK IF BMP.
2920 020553 001002 ; BIC R2,R0 ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
2921 020554 004767 001470 ; BNE 4$ ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
2922 ; JSR PC,SAVBMP ;SAVE BMP CODE ON THE QUEUE.
2923 020555 005301 4$: DEC R1 ;DECREMENT THE TRY COUNT.
2924 020556 001362 ; BNE 2$ ;LOOP TO TRY AGAIN.
2925 020557 000241 ; CLC ;CLEAR CARRY, TO INDICATE FIFO NOT PURGED.
2926 020558 000401 ; BR 60$ ;EXIT WITH CARRY CLEAR.
2927 020559 000261 6$: SEC ;SET CARRY, TO INDICATE FIFO PURGED.
2928 ;
2929 020560 004736 60$: PASS ;RESTORE GPRS,
2930 ; JSR PC,0(SP) ;RETURN TO PREG05 SUBRT.
2931 020561 000207 RTS PC ;CARRY BIT, SET INDICATES FIFO PURGED.

```

N6

CZOHUBO DMU 11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 64
GLOBAL SUBROUTINE - RDPDR

SEQ 78

```

2933 .SBTTL GLOBAL SUBROUTINE - RDPDR -
2934 *****
2935 * - READ AND VERIFY DATA PATTERN FROM DEVICE REGISTERS ROUTINE
2936 * THIS ROUTINE READS AND VERIFIES THE ROTATED DATA PATTERN WHICH HAS
2937 * BEEN WRITTEN BY THE WDPDR SUBROUTINE.
2938 * EACH ACTIVE LINE'S REGISTER'S CONTENTS IS READ AND COMPARED WITH THE
2939 * WRITTEN DATA.
2940 * AFTER THE UNUSED AND READ ONLY (RO) BITS ARE MASKED OUT, ANY ERRORS ARE
2941 * REPORTED FROM THIS ROUTINE.
2942 * THIS ROUTINE WILL TAKE INTO ACCOUNT THE TYPE OF WRITE OPERATION WHICH
2943 * WAS PERFORMED BY THE WDPDR SUBROUTINE.
2944 *
2945 * INPUTS: R2 - USED TO PASS IN THE DATA PATTERN TO BE ROTATED & VERIFIED.
2946 * R3 - BYTE INDICATOR (- > LO BYTE, + > HI BYTE, 0 > BOTH).
2947 * R4 - OPERATION TYPE INDICATOR (- > BIC, + > BIS, 0 > MOV).
2948 * ACTLNS - BIT MAP OF ACTIVE LINES ON THE DEVICE UNDER TEST.
2949 * CSRA - CONTAINS THE CSR ADDRESS OF THE DEVICE UNDER TEST.
2950 * DRADRT - BASE ADDRESS OF DEVICE REGISTER ADDRESS TABLE.
2951 * ERCNTB - LABEL AT BASE OF ERROR COUNTERS TABLE FOR LINES.
2952 * ERRMSG - SET UP WITH THE PROPER ERROR MESSAGE FOR THIS TEST.
2953 * ERRNBR - SET UP WITH THE PROPER ERROR NUMBER.
2954 * LPRO - EQUATED TO LPR REG OFFSET FROM DEVICE CSR ADDRESS.
2955 * NUMLNS - NUMBER OF LINES ON THE DEVICE UNDER TEST.
2956 * NDERPT - NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE.
2957 * TXBFCO - EQUATED TO TBUFCY REG OFFSET FROM DEVICE CSR ADDRESS.
2958 * UNBTB - BASE ADDRESS OF THE UNUSED BIT TABLE.
2959 *
2960 * OUTPUTS: ERROR MESSAGES MAY BE PRINTED AT THE OPERATOR'S CONSOLE.
2961 * ERCNT - ERROR COUNTERS TABLE IS UPDATED FOR LINE UNDER TEST.
2962 * ERRBLK - CONTENTS DESTROYED.
2963 * ERSRMR - ERROR SUMMARY FLAGS BIT SET IF LINE IN SUMMARY MODE.
2964 * UUT CSR - ALL BITS CLEARED, EXCEPT IND.ADR.REG FIELD DESTROYED.
2965 *
2966 * CALLING SEQUENCE: JSR PC,RDPDR
2967 *
2968 * COMMENTS: FOR BYTE ACCESSES, ONLY THE SPECIFIED BYTE IS VERIFIED.
2969 *
2970 * SUBORDINATE ROUTINES CALLED: ER1601,ROLDAP.
2971 * - *****
2972
2973 020624 RDPDR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2974 020624 004567 163146 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2975 020630 012767 016416 163136 MOV #ER1601,ERRBLK ;SET UP THE ADDRESS OF THE EPROR REPORT RTN.
2976 *
2977 * DETERMINE WHETHER REGISTER DATA SHOULD BE INVERTED FROM DATA PATTERN.
2978 *
2979 020636 005704 TST R4 ;CHECK THE OPERAND TYPE INDICATOR.
2980 020640 100001 BPL R5 ;BIC WRITE PERFORMED? NO, USE STANDARD DATA.
2981 020642 005102 COM R2 ;YES, INVERT THE DATA PATTERN.
2982 *
2983 * SET UP OUTER LOOP.
2984 020644 005005 CLR R5 ;CLEAR LINE COUNTER TO SELECT LINE 0.
2985 *
2986 * THE OUTER LOOP FOLLOWS. EACH PASS THROUGH THIS LOOP READS AND COMPARES DATA
2987 * FROM ALL OF THE DEVICE REGISTERS FOR A PARTICULAR LINE IF THE LINE IS ACTIVE.
2988 *

```



```

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

```

GLOBAL SUBROUTINE MACRO M1200 15-MAR 84 09:15 PAGE 64-2

ADDRESS	HEX	ASCII	OPERATION	COMMENT
3045	021050	016702	000020	16\$: MOV 70\$,R2 ;SET UP TO ROTATE THE DATA PATTERN.
3046	021054	004767	000410	JSR PC,ROLDAP ;ROTATE THE DATA PATTERN.
3047	021060	005205		INC R5 ;COUNT THIS LINE.
3048	021062	020527	000020	CMP R5,#NUMLNS ;COMPARE LINE COUNT WITH NUMBER OF LINES.
3049	021066	002667		BLT 4\$;LOOP IF SOME LINES NOT DONE.
3050				
3051	021070			60\$: PASS ;RESTORE GPRS.
3052	021070	004736		JSR PC,8(SP). ;RETURN TO PREG05 SUBRT.
3053	021072	000207		RTS PC
3054	021074	000000		70\$: .WORD 0 ;STORAGE FOR DATA PATTERN OUTSIDE INNER LOOP.

D7

GLOBAL SUBROUTINE

```

3056 .SBTTL GLOBAL SUBROUTINE - REGTST -
3057 *** *****
3058 - REGISTERS TEST SUBROUTINE -
3059 SUBROUTINE TO TEST THE DEVICE UNDER TEST (DUT) REGISTERS. THE USED
3060 BITS OF THE REGISTERS ARE EITHER ALL CLEARED OR ALL SET AND THEN THE
3061 DATA PATTERN IS WRITTEN AND VERIFIED USING EITHER WORD OR BYTE
3062 ACCESSES IN READ/WRITE OR READ/MODIFY/WRITE MODE.
3063
3064 INPUTS: R3 - BYTE INDICATOR (- > LOW, + > HIGH, 0 > BOTH BYTES).
3065 R4 - ACCESS MODE (-1 > SET THEN BIC, 1 > CLEAR THEN BIS,
3066 (-2 > SET THEN MOV, +2 CLEAR THEN MOV).
3067 ERRNBR - SET UP WITH INITIAL ERROR NUMBER.
3068
3069 OUTPUTS: GPRS0 - GPR SAVE AREA 0 IS DESTROYED.
3070 DEVICE UNDER TEST REGISTER'S ARE WRITTEN.
3071 ERROR MESSAGES MAY BE PRINTED AT THE OPERATORS CONSOLE.
3072
3073 CALLING SEQUENCE: JSR PC,REGTST
3074
3075 COMMENTS: THIS ROUTINE LOOP 16 TIMES WRITING THE SAME DATA PATTERN
3076 ROTATED LEFT ONCE EACH ITERATION.
3077 THIS ROUTINE CAN REPORT ERRORS INITIAL ERRNBR THRU INITIAL+2.
3078
3079 SUBORDINATE ROUTINES CALLED: RDPDR,ROLDAP,SWAPO,WOPDR
3080
3081
3082 021076 REGTST:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
3083 021076 004567 162674 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3084
3085 ; SET UP THE GPRS FOR THE WRITING OF THE DATA PATTERN.
3086
3087 021102 012705 000020 MOV #16,R5 ;SET UP LOOP COUNTER TO COUNT 16 ITERATIONS.
3088 021106 012702 167410 MOV #167410,R2 ;INITIALIZE THE DATA PATTERN.
3089 021112 032704 000001 BIT #0,R4 ;TEST FOR R/W ACCESS.
3090 021116 001001 BNE 2$ ;R/M/W ACCESS? YES, R4 IS ALL SET UP.
3091 021122 005004 CLR R4 ;NO, INDICATE R/W ACCESS.
3092
3093 2$:
3094
3095 ; SET UP THE GPRS FOR THE CLEARING OR SETTING OF ALL THE USED BITS.
3096
3097 021122 010400 MOV R4,R0 ;PASS OPERATION TYPE INDICATOR AROUND SWAPO.
3098 021124 004767 001272 JSR PC,SWAPO ;GET ALTERNATE GPR SET IN R1 THRU R5.
3099 021130 016701 162634 MOV ERRNBR,R1 ;SAVE THE INITIAL ERROR NUMBER.
3100 021134 010004 MOV R0,R4
3101 021136 005404 NEG R4 ;SET UP OP TYPE FOR CLEARING OR SETTING.
3102 021140 005002 CLR R2 ;SET UP CLEAR WRITE PATTERN.
3103 021142 026627 000012 000002 CMP R4,SLOT(SP),#2 ;TEST FOR CLEAR THEN MOV TEST SEQUENCE.
3104 021150 001401 BEQ 4$ ;CLEAR THEN MOV? YES, LEAVE WRITE PAT CLEAR.
3105 021152 005102 COM R2 ;NO, SET ALL BITS OF WRITE PATTERN.
3106 021154 005000 CLR R3 ;INDICATE THAT WORD ACCESSES SHOULD BE USED.
3107 021160 026627 000012 177776 CMP R4,SLOT(SP),#-2 ;SET ALTERNATE BYTE EXPECTED DATA PAT TO CLEAR.
3108 021166 001001 BNE 6$ ;TEST FOR SET THEN MOV TEST SEQUENCE.
3109 021170 005100 COM R0 ;SET THEN MOV? YES, LEAVE ALT BYTE PAT CLEAR.
3110 021172 004767 001224 JSR PC,SWAPO ;NO, SET ALT BYTE EXPECTED DATA PAT TO ALL 1'S.
3111 ;RESTORE SWAPPED GPR VALUES TO R1 THRU R5.
3112
3113 ; START OF DATA PATTERN LOOP.

```

E7

CZDHUBO DHU-11 FUNC TST PART1
GLOBAL SUBROUTINEMACRO M1200 15-MAR-84 09:15 PAGE 65-1
- REGTST -

SEQ 82

```

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

```

F7

CZDHUBO DHU-11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 65-2
GLOBAL SUBROUTINE - REGIST

SEQ 83

```
3169  
3170 021316 016767 161126 162444 60$: MOV GPRS00,ERRNBR ;YES, RESTORE ERROR NUMBER AND EXIT.  
3171 021324 PASS ;GET THE ERROR NUMBR FROM GPR SWAP STORAGE.  
021324 004736 ;RESTORE GPRS.  
3172 021326 000207 RTS PC JSR PC,0(SP)+ ;RETURN TO F1EG05 SUBRT.
```

67

CZDHUBO DHU-11 FUNC TST PART1
GLOBAL SUBROUTINE

MACRO M1200 15-MAR-84 09:15 PAGE 66
- REPSMR -

SEQ 84

```

3174 .SBTTL GLOBAL SUBROUTINE - REPSMR -
3175 ;* *****
3176 ;* - REPORT ERROR SUMMARY ROUTINE -
3177 ;* THIS SUBROUTINE REPORTS AN ERROR SUMMARY FOR THOSE LINES WHICH HAVE
3178 ;* EXCEEDED THE NUMBER OF INDIVIDUAL ERRORS TO REPORT FOR A SINGLE LINE
3179 ;* IN A SINGLE TEST. THIS PARAMETER CAN BE SPECIFIED BY THE OPERATOR IF
3180 ;* HE/SHE ANSWERS THE SOFTWARE PARAMETER QUESTIONS.
3181 ;*
3182 ;* INPUTS: ERNTB - LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
3183 ;* ERRMSG - ADDRESS OF PRIMARY ERROR MESSAGE.
3184 ;* ERRNBR - ERROR NUMBER OF ERRORS IN THIS ROUTINE.
3185 ;* ERSMBF - "REPORT ERROR SUMMARY FOR LINE" FLAGS.
3186 ;*
3187 ;* OUTPUTS: ERRBLK - ADDRESS OF ERROR REPORTING ROUTINE (DESTROYED).
3188 ;* SUMMARY MESSAGES MAY BE PRINTED AT THE OPERATOR CONSOLE.
3189 ;*
3190 ;* CALLING SEQUENCE: JSR PC,REPSMR
3191 ;*
3192 ;* COMMENTS: IF NO LINES HAVE EXCEEDED THE MAXIMUM NUMBER OF INDIVIDUAL
3193 ;* ERRORS TO REPORT, NO MESSAGES ARE PRINTED BY THIS ROUTINE.
3194 ;* ERROR SUMMARIES IN THIS ROUTINE ARE REPORTED AS ERRORS.
3195 ;* THE CONTENTS OF ERRBLK ARE DESTROYED.
3196 ;*
3197 ;* SUBORDINATE ROUTINES CALLED:
3198 ;* - *****
3199
3200 021330 REPSMR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3201 021330 004567 162442 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3202 021334 005767 161122 TST ERSMBF ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
3203 BEQ 60$ ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
3204
3205 ;* WE HAVE SOME ERROR SUMMARIES TO REPORT.
3206 021342 012767 016722 162424 MOV #ER9004,ERRBLK ;SELECT ERROR REPORTING ROUTINE.
3207 ;*
3208 ;* REPORT
3209 ;* "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
3210 ;*
3211 021350 ERROR
3212 021350 104460 TRAP C$ERROR
3213 021352 004736 60$: PASS ;RESTORE GPRS.
3214 021354 000207 RTS PC JSR PC,@(SP), ;RETURN TO PREG05 SUBRT.

```

H/

CZDHUBO DHU-11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 67
GLOBAL SUBROUTINE . RESETT

SEQ 85

```

3216 .SRTTL GLOBAL SUBROUTINE . - RESETT -
3217 *****
3218 ; - RESET DEVICE UNDER TEST -
3219 ; THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
3220 ; IF RESET DOES NOT SUCCESSFULLY COMPLETE, IE. TIME-OUT OCCURS, THEN
3221 ; AN ABORT TEST ERROR MESSAGE IS REPORTED.
3222 ;
3223 ; INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR
3224 ; TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
3225 ; ERRIBL - ERRITYP,EPNBR,AND ERRMSG SET UP CORRECTLY.
3226 ;
3227 ; OUTPUTS: THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
3228 ; CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
3229 ; ERRBLK - VALUE MAY BE DESTROYED.
3230 ; IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
3231 ; TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
3232 ;
3233 ; CALLING SEQUENCE: JSR PC,RESETT
3234 ;
3235 ; COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERRNBR
3236 ; THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
3237 ;
3238 ; SUBORDINATE ROUTINES CALLED: DELAY,MSLGET.
3239 *****
3240
3241 021356 RESETT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3242 021356 004567 162414 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3243 021362 012702 000040 MOV #BIT05,R2 ;SET BIT MASK OF MASTER RESET BIT.
3244 ;
3245 ; TEST THE STATE OF THE MASTER RESET BIT IN THE CSR.
3246 ; IF MR IS SET THEN WAIT FOR SELF-TEST TO COMPLETE.
3247 ; IF TIME-OUT OCCURS, REPORT THE ERROR AND PASS-OUT ABORT TEST INDICATOR.
3248 021366 016704 160650 MOV CSRA,R4 ;GET THE ADDRESS OF THE DUT'S CSR.
3249 021372 030214 BIT R2,(R4) ;CHECK STATE OF MASTER RESET BIT.
3250 021374 001406 BFG 2$ ;DON'T DELAY IF MR IS ALREADY CLEAR.
3251 021376 005003 CLR R3 ;SET UP DESIRED STATE OF MASTER RESET BIT.
3252 021400 012701 011610 MOV #5000.,R1 ;PASS TIME-OUT VALUE OF 5 SECONDS.
3253 021404 004767 176554 JSR PC,MSLGET ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
3254 021410 103012 BCC 4$ ;GO REPORT ERROR IF TIMEOUT OCCURRED.
3255 ;
3256 ;
3257 ; SET MASTER RESET BIT IN CSR. CLEAR TX AND RX ENABLE BITS, ETC.
3258 ; SKIP THE SELFTEST.
3259 ; TIME-OUT OF 5 SECS. JUST IN CASE THE SELF-TEST EXECUTES.
3260 ;
3261 021412 010277 160624 2$: MOV R2,0CSRA ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
3262 021416 004767 000722 JSR PC,SKPSTS ;TRY TO SKIP THE SELFTEST.
3263 ;
3264 ; SET SELF TEST TIME OUT OF 5 SECONDS, AND WAIT FOR M.R TO CLEAR.
3265 ; IF TIME OUT OCCURS, THEN REPORT THE FATAL ERROR AND PASS-OUT THE ABORT
3266 ; TEST INDICATOR.
3267 ;
3268 021422 005003 CLR R3 ;SET UP DESIRED STATE OF MASTER RESET BIT.
3269 021424 012701 011610 MOV #5000.,R1 ;PASS TIME-OUT VALUE OF 5 SECONDS.
3270 021430 004767 176530 JSR PC,MSLGET ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
3271 021434 103410 BCS 6$ ;SKIP ERROR REPORT IF MR CLEARED IN TIME.

```

```

3272      ;
3273      ; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET, TEST ABORTED".
3274      ; INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
3275      ;
3276 021436 012701 012221 4$: MOV    #EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
3277 021442 012767 016526 162324 MOV    #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
3278      ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
3279      ; "TEST ABORTED"
3280 021450      ERROR      ;          >>>>> ERROR <<<<<
3281      021450 104460      TRAP    C$ERROR
3282 021452 000241      CLC          ;INDICATE TEST IS TO BE ABORTED.
3283      021454 000403      BR       60$      ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
3284      ;
3285      ; CLEAR TX AND RX INTERRUPT ENABLE STATUS FLAGS IN IESTAT.
3286      ; EXIT WITH CONTINUE TEST INDICATOR SET (IE,CARRY SET).
3287 021456 005067 160626 6$: CLR     IESTAT      ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
3288 021462 000261      SEC          ;INDICATE SUCCESS, CONTINUE TEST.
3289      ;
3290 021464      60$: PASS          ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
3291      021464 004736      JSR      PC,@(SP)+    ;RETURN TO PREGOS SUBRT.
3292 021466 000207      RTS     PC      ;CARRY BIT; IF CLEAR, INDICATES ABORT TEST.
3293

```



```
3295 .SHTTL GLOBAL SUBROUTINE - ROLDAP -
3296 ;*****
3297 ;* - ROTATE LEFT DATA PATTERN
3298 ;* THIS ROUTINE ROTATES THE PASSED INPUT DATA PATTERN LEFT,WITHOUT GOING
3299 ;* THROUGH THE CARRY,THE CARRY IS INITIALLY SET OR CLEARED DEPENDING
3300 ;* UPON THE STATE OF THE MSB OF THE DATA PATTERN BEFORE A ROL INSTRUCTION
3301 ;* IS EXECUTED.
3302 ;*
3303 ;* INPUTS: R2 - CONTAINS THE DATA PATTERN TO BE ROTATED
3304 ;*
3305 ;* OUTPUTS: R2 - CONTAINS THE ROTATED DATA PATTERN
3306 ;*
3307 ;* CALLING SEQUENCE: JSR PC,ROLDAP
3308 ;*
3309 ;* COMMENTS:
3310 ;*
3311 ;* SUBORDINATE ROUTINES CALLED: NONE
3312 ;*****
3313
3314 021470 ROLDAP::SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3315 021470 004567 162302 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3316 021474 005702 ;CHECK MSB, AND CLEAR CARRY.
3317 021476 100001 ;BRANCH IF CLEAR.
3318 021500 000261 ;SET CARRY IF MSB SET
3319 021502 006102 ;ROTATE DATA PATTERN LEFT
3320 021504 010266 000006 ;RESTORE GPRS,EXCEPT
3321 021510 004736 MOV R2,R2SLOT(SP) ;PUT R2 IN STACK SLOT.
3321 021512 000207 JSR PC,@(SP). ;RETURN TO PREG05 SUBRT.
;R2 - CONTAINS THE ROTATED DATA PATTERN
RTS PC
```

```

SBT11 GLOBAL SUBROUTINE - RSTRPT -
*****
- REPORT ANY RESET ERRORS ROUTINE
THIS ROUTINE DETERMINES IF ANY ERROR CODES ARE AMONG THE DIAGNOSTIC
CODES REPORTED PLACED IN THE DUT RECEIVED CHARACTER FIFO BY THE
SELF-TEST. IF ANY NON BMP ERROR CODES ARE FOUND, OR IF OTHER ERRORS
ARE ENCOUNTERED, APPROPRIATE ERRORS ARE REPORTED. ANY BMP CODES THAT
ARE FOUND, ARE PLACED ON THE BMP CODE QUEUE TO BE REPORTED LATER.
THIS ROUTINE ALSO PURGES THE DUT FIFO LOOKING FOR ANY CHARACTERS
OR MODEM STATUS CODES. IF ANY ARE FOUND, ERRORS ARE REPORTED.

INPUTS:  ERRMSG - ADDRESS OF THE PRIMARY ERROR MESSAGE.
         ERRNBR - ERROR NUMBER OF FIRST ERROR REPORTED BY THIS ROUTINE.
         NUMLNS - EQUATED TO THE NUMBER OF LINE ON THE DUT.
         RBUFA - CONTAINS ADDRESS OF THE DUT RECEIVER FIFO.

OUTPUTS:  CARRY - SUCCESS FLAG (SET IF FIFO CLEARED SUCCESSFULLY).
         ERRBLK - ADDRESS OF THE ERROR REPORT ROUTINE (DESTROYED).
         ERROR MESSAGES CAN BE PRINTED AT THE OPERATORS CONSOLE.

CALLING SEQUENCE:  JSR      PC,RSTRPT

COMMENTS:  THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERRNBR
           THRU INITIAL ERRNBR+4.
           THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.

SUBORDINATE ROUTINES CALLED: ER0503,ER9007,ER9008,SAVBMP.
*****

RSTRPT:: SAVE                ;SAVE CONTENTS OF GPRS R0 THRU R5.
                        JSR      R5,PREG05          ;CALL REGISTER SAVE SUBRT.

; READ CORRECT NUMBER (NUMBER OF LINE ON DUT) OF CHARS FROM THE FIFO.
; VERIFY THAT EACH CHAR IS A SELFTEST SUCCESS CODE.
;
      CLR      R3                ;CLEAR THE CODE COUNTER.
      MOV      ERRNBR,R5          ;SAVE ERRNBR FOR RESTORATION LATER.
2$:  MOV      @RBUFA,R2          ;READ A CHAR FROM THE DUT FIFO.
      BMI     4$                ;SKIP ERROR IF DATA VALID SET FOR CHAR.
; WE EXPECT A SELFTEST CODE, BUT THIS FIFO SLOT IS EMPTY.
;
      MOV      R5,ERRNBR          ;RESTORE ERROR NUMBER TO INITIAL VALUE.
      MOV      @EM9018,R1         ;PASS ERROR MESSAGE IN#0 TO ER9007 ROUTINE.
      MOV      @ER9007,ERRBLK     ;SELECT PROPER ERROR REPORT ROUTINE.
; REPORT ERROR WITH NUMBER INITIAL ERRNBR.
; NO SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET.
;
      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     3$                ;AVOID SET THE FLAG IF IT HAS.
      MOV      @1,EX0ERR         ;SET THE EXIT ON ERROR FLAG

```



```

3378
3379
3380
3381 021572 000261
3382 021574 000167 000406
3383
3384
3385
3386 021600 012700 070001
3387 021604 040200
3388 021606 001042
3389
3390
3391
3392
3393 021610 032702 000200
3394 021614 001462
3395 021616 120227 000203
3396 021622 001457
3397 021624 120227 000201
3398 021630 001454
3399 021632 012700 000300
3400 021636 040200
3401 021640 001003
3402 021642 004767 000430
3403 021646 000445
3404
3405
3406
3407 021650 010567 162114
3408 021654 005267 162110
3409 021660 012701 015154
3410 021664 012767 017126 162102
3411
3412
3413
3414
3415 021672
    021672 104460
3416
3417
3418
3419 021674 032767 000100 160322
3420 021702 001027
3421
3422 021704 012767 000001 160372
3423 021712 000534
3424
3425
3426
3427
3428 021714 010567 162050
3429 021720 062767 000002 162042
3430 021726 012701 015137
3431 021732 012767 017036 162034
3432
3433

;+
; 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.
;-

```

```

3434      ; "NON-SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET."
3435      ;
3436 021740      ; ERROR                                ; >>>> ERROR <<<<.
                                ; TRAP      C$ERROR
3437 021740      ;
3438      ; EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3439      ;
3440 021742      ; 032767      000100      160254      ; BIT      #BIT06,OPTION      ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3441 021750      ; 001004      ; BNE      10$      ;AVOID SET THE FLAG IF IT HAS AND GO TO
3442      ;                               ;THE END OF THE LOOP.
3443 021752      ; 012767      000001      160324      ; MOV      #1,EXOERR      ;SET THE "EXIT ON ERROR" FLAG
3444 021760      ; 000511      ; BR      50$      ;EXIT THE ROUTINE WITH FAILURE.
3445      ;
3446      ;
3447      ; END OF LOOP, LOOP IF NOT ALL CHARS HAVE BEEN READ FROM THE FIFO.
3448      ;
3449 021762      ; 005203      10$:      INC      R3      ;SET CODE COUNTER FOR NEXT ITERATION OF LOOP.
3450 021764      ; 020327      000010      ; CMP      R3,#8.      ;TEST FOR ALL CODES READ.
3451 021770      ; 002656      ; BLT      2$      ;LOOP IF NOT CHARS READ FROM FIFO.
3452      ;
3453      ; PURGE THE FIFO UNTIL DATA.VALID IS CLEAR OR UNTIL TOO MANY CHARS ARE READ.
3454      ;
3455 021772      ; 012704      000022      ; MOV      #18,,R4      ;INITIALIZE THE CHARACTER COUNTER.
3456 021776      ; 010567      161766      ; MOV      R5,ERRNBR      ;GET INITIAL VALUE OF THE ERROR NUMBER.
3457 022002      ; 062767      000003      161760      ; ADD      #3,ERRNBR      ;CALCULATE ERROR NUMBER OF NEXT ERROR.
3458 022010      ; 012767      017126      161756      ; MOV      #ER9008,ERRBLK      ;SELECT PROPER ERROR REPORT ROUTINE.
3459 022016      ; 017702      160222      12$:      MOV      @RBUFA,R2      ;READ A CHARACTER FROM THE OUT FIFO.
3460 022022      ; 000261      ; SEC      ;INDICATE SUCCESS IN CASE DATA.VALID IS CLEAR.
3461 022024      ; 100070      ; BPL      60$      ;EXIT ROUTINE WITH SUCCESS IF DATA.VALID CLEAR.
3462      ;
3463      ; WE HAVE A CHARACTER.
3464      ; DETERMINE IF CHAPACTER IS A DATA CHARACTER.
3465      ;
3466 022026      ; 012700      070000      ; MOV      #70000,R0      ;TEST BITS 12 THRU 14 OF THE
3467 022032      ; 040200      ; BIC      R2,R0      ; CODE READ FROM THE OUT FIFO.
3468 022034      ; 001403      ; BEQ      14$      ;SKIP THIS ERROR IF CODE IS NOT A DATA CHAR.
3469      ;
3470      ; WE HAVE AN UNEXPECTED DATA CHARACTER; SET UP AND GO TO REPORT ERROR.
3471      ;
3472 022036      ; 012701      015200      ; MOV      #EM9022,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3473 022042      ; 000423      ; BR      22$      ;GO TO REPORT THIS ERROR.
3474      ;
3475      ; WE HAVE AN UNEXPECTED CODE.
3476      ; DETERMINE IF THE CODE IS A MODEM STATUS CODE.
3477      ;
3478 022044      ; 032702      000001      14$:      BIT      #BIT0,R2      ;TEST MODEM STATUS INDICATOR BIT OF CODE.
3479 022050      ; 001003      ; BNE      16$      ;SKIP THIS ERROR IF NOT MODEM STATUS CODE.
3480      ;
3481      ; WE HAVE A MODEM STATUS CODE; SET UP AND GO TO REPORT ERROR.
3482      ;
3483 022052      ; 012701      015217      ; MOV      #EM9023,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3484 022056      ; 000415      ; BR      22$      ;GO TO REPORT THIS ERROR.
3485      ;
3486      ; WE HAVE AN ONBOARD TEST CODE.
3487      ; DETERMINE IF THIS CODE IS A BMP CODE.
3488      ;
3489 022060      ; 032702      000200      16$:      BIT      #BIT7,R2      ;TEST THE ROM VERSION BIT OF THE CODE.

```

N7

CZDH080 DHU-11 FUNC TST PART1 MACRO M1200 15-MAR 84 09:15 PAGE 69-3
GLOBAL SUBROUTINE - RSTRPT -

SEQ 91

```

3490 022064 001404          BEQ      18$          ;GOTO SET UP FOR SELFTEST CODE IF ROM VERSION.
3491 022066 012700 000300    MOV      $300,R0
3492 022072 040200          BIC      R2,R0          ;TEST THE ERROR TYPE BITS OF THE CODE.
3493 022074 001403          BEQ      20$          ;SKIP THIS ERROR IF BMP CODE.
3494
3495          ;* WE HAVE A SELFTEST CODE: SET UP AND GO TO REPORT ERROR.
3496          ;*
3497 022076 012701 015241    18$:    MOV      $EM9024,R1          ;SELECT ERROR MSG INFO FOR ER0806 ROUTINE.
3498 022102 000403          BR       22$          ;GO TO REPORT THIS ERROR.
3499
3500          ;* WE HAVE A BMP CODE: SAVE IT ON THE QUEUE.
3501          ;*
3502 022104 004767 000166    20$:    JSR      PC,SAVBMP          ;SAVE THE BMP CODE ON THE QUEUE.
3503 022110 000411          BR       24$          ;
3504
3505          ;* REPORT THE ERROR WITH ERROR NUMBER OF INITIAL ERRNBR + 3.
3506          ;* "UNEXPECTED XXX XXXX FOR LINE NM IN FIFO AFTER RESET:"
3507          ;*
3508 022112          22$:    ERROR          ;          >>>> ERROR <<<<<
3509 022112 104460          TRAP      C$ERROR
3510
3511          ;* EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3512          ;*
3512 022114 032767 000100 160102    BIT      $BIT06,OPTION          ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3513 022122 001004          BNE      24$          ;AVOID SETTING THE FLAG IF IT HAS AND GO TO
3514          ;* THE END OF THE LOOP.
3515 022124 012767 000001 160152    MOV      $1,EXOERR          ;SET THE "EXIT ON ERROR" FLAG
3516 022132 000424          BR       50$          ;EXIT THE ROUTINE WITH FAILURE.
3517
3518          ;*
3519          ;* END OF LOOP.
3520          ;* COUNT THE CHARACTER WE JUST RECEIVED, AND CHECK FOR TOO MANY RECEIVED.
3521          ;*
3522 022134 005304          24$:    DEC      R4          ;COUNT THIS CHARACTER.
3523 022136 001327          BNE      12$          ;LOOP IF NOT TOO MANY CHARACTERS PURGED.
3524
3525          ;* WE READ TOO MANY VALID CHARACTERS WHILE TRYING TO PURGE THE FIFO.
3526          ;* REPORT ERROR AND EXIT WITHOUT SUCCESS.
3527          ;* "FIFO WILL NOT PURGE (DATA VALID STUCK SET), REMAINDER OF TEST SKIPPED."
3528          ;*
3529 022140 012701 015016          MOV      $EM9017,R1          ;SELECT PROPER ERROR MESSAGE.
3530 022144 010567 161620          MOV      R5,ERRNBR          ;GET INITIAL ERROR NUMBER.
3531 022150 062767 000004 161612    ADD      $4,ERRNBR          ;CALCULATE INITIAL ERRNBR + 4.
3532 022156 012767 016136 161610    MOV      $ERC503,ERRBLK          ;SELECT PROPER ERROR REPORT ROUTINE.
3533          ;PRINT ERROR REPORT.
3534 022164          ERROR          ;          >>>> ERROR <<<<<
3535 022164 104460          TRAP      C$ERROR
3536
3537          ;* EXIT THIS ROUTINE IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
3538          ;*
3538 022166 032767 000100 160030    BIT      $BIT06,OPTION          ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
3539 022174 001003          BNE      50$          ;AVOID SETTING THE FLAG IF IT HAS.
3540 022176 012767 000001 160100    MOV      $1,EXOERR          ;SET THE "EXIT ON ERROR" FLAG
3541
3542 022204 000241          50$:    CLC          ;CLEAR THE SUCCESS FLAG.
3543
3544 022206          60$:    PASS          ;RESTORE GPRS,

```

B8

C:\DPM\NO DMU 11 FUNC TST PART1 MACRO M1200 15-MAR 84 09:15 PAGE 69-4
GLOBAL SUBROUTINE - RSTRT -

SEQ 92

0.00006 004.36
3545 022210 000207

RTS PC

JSR

PC,0(SP);
1 CARRY - SUCCESS FLAG (SET IF FIFO IS PURGED).

;RETURN TO PREG05 SUBRT.

C8

CZDHU10 DHU 11 LUNC TST PART1 MACHO M1200 15 MAR 84 09:15 PAGE 70

SEQ 93

```

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

```

D8

CZDHUB0.DHU 11 LUNC 1ST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 71
GLOBAL SUBROUTINE - RXIE1

SEQ 94

```

3574 .SBTTL GLOBAL SUBROUTINE - RXIE1
3575 ;*****
3576 ;* RECEIVER INTERRUPT ENABLE
3577 ;* THIS ROUTINE IS USED TO ENABLE RECEIVER INTERRUPTS IN THE DHU11.
3578 ;*
3579 ;* INPUTS: NONE.
3580 ;*
3581 ;* OUTPUTS: THE RX.INT.ENBL BIT IS SET IN THE OUT CSR.
3582 ;* IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3583 ;* ENABLE BITS.
3584 ;*
3585 ;* CALLING SEQUENCE: JSR PC,RXIE1
3586 ;*
3587 ;* COMMENTS: THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3588 ;* THE OUT CSR ARE DESTROYED.
3589 ;*
3590 ;* SUBORDINATE ROUTINES CALLED: NONE.
3591 ;*****
3592
3593 022252 052767 000100 160030 RXIE1:: BIS 081706,IESTAT ;SET RX.INT.ENBL BIT IN IESTAT.
3594 022260 042767 137677 160022 BIC 0137677,IESTAT ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
3595 022266 016777 160016 157746 MOV IESTAT,0CSRA ;ENABLE RX INTERRUPTS.
3596 022274 000207 RTS PC

```



```

3598 .SBTTL GLOBAL SUBROUTINE - SAVBMP -
3599 ;* *****
3600 ;* - SAVE BMP CODES ROUTINE -
3601 ;* THIS ROUTINE SAVES THE PARAMETER PASSED IN, ONTO THE BMP CODE QUEUE
3602 ;* TOGETHER WITH THE NUMBER OF THE CURRENTLY EXECUTING TEST.
3603 ;*
3604 ;* INPUTS: R2 - CONTAINS THE BMP CODE THAT IS TO BE PLACED ON THE QUEUE.
3605 ;* BMPCQP - CONTAINS ADDRESS OF NEXT LOCATION IN THE BMP QUEUE.
3606 ;* BMPCQB - LABEL AT BASE OF THE BMP CODE QUEUE.
3607 ;* BMPCQE - LABEL OF NEXT LOCATION AFTER THE END OF THE BMP QUEUE.
3608 ;* TSTNUM - CONTAINS THE NUMBER OF THE CURRENT TEST.
3609 ;*
3610 ;* OUTPUTS: BMPCQP - INCREMENTED BY 4.
3611 ;* THE CONTENTS OF THE BMP CODE QUEUE ARE UPDATED.
3612 ;*
3613 ;* CALLING SEQUENCE: JSR PC,SAVBMP
3614 ;*
3615 ;* COMMENTS: IF THE OVERFLOW OCCURS THEN THE LAST LOCATION WILL BE
3616 ;* OVERWRITTEN BY ANY SUBSEQUENT ATTEMPTS TO UPDATE THE QUEUE.
3617 ;*
3618 ;* SUBORDINATE ROUTINES CALLED: NONE.
3619 ;* - *****
3620
3621 022276 SAVBMP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3622 022276 004567 161474 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3623 022302 016704 160216 ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
3624 022306 116724 160012 ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
3625 022312 005204 ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
3626 022314 042702 177400 ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
3627 022320 010224 ;SAVE THE BMP CODE ON THE QUEUE.
3628 022322 020427 002726 ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
3629 022326 103402 ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
3630 022330 162704 000004 ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
3631 022334 010467 160164 ;SAVE THE POINTER.
3632 022340 ;RESTORE GPRS.
3633 022340 004736 ;PC,(SP)+ ;RETURN TO PREG05 SUBRT.
3633 022342 000207 RTS PC JSR

```

```

3635 .SBTTL GLOBAL SUBROUTINE - SKPSTS -
3636 ;+ *****
3637 ;* - SKIP SELFTEST ROUTINE -
3638 ;* THIS SUBROUTINE IS USED TO SKIP THE SELFTEST AFTER A DUT RESET HAS BEEN
3639 ;* INITIATED. IT MUST BE ENTERED IMMEDIATELY AFTER SETTING THE DUT MASTER
3640 ;* RESET ROUTINE OR AFTER THE EXECUTION OF A BUS RESET (BECAUSE OF TIMING
3641 ;* CONSIDERATIONS).
3642 ;*
3643 ;* INPUTS: CSRA CONTAINS ADDRESS OF THE DUT CSR.
3644 ;* TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
3645 ;*
3646 ;* OUTPUTS: SKIP SELFTEST CODES ARE WRITTEN TO THE DUT REGISTERS.
3647 ;*
3648 ;* CALLING SEQUENCE: JSR PC,SKPSTS
3649 ;*
3650 ;* COMMENTS:
3651 ;*
3652 ;* SUBORDINATE ROUTINES CALLED: DELAY.
3653 ;- *****
3654
3655 022344 SKPSTS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3656 022344 004567 161426 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3657 022350 012704 000012 MOV #10,,R4 ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
3658 022354 004767 175544 JSR PC,DELAY ;DELAY FOR 10 MILLI-SECONDS.
3659
3660 ;+
3661 022360 012701 000060 ; WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
3662 ;-
3663 MOV #NUMLNS:BIT05,R1 ;FORM IND.ADR.REG FIELD (PLUS M.R. BIT) WORD.
3664 022364 012703 052525 ;THE ABOVE INCLUSION OF THE M.R. BIT IS NECESSARY BECAUSE OF THE
3665 022370 005301 ; LACK OF A M.R. BIT WRITE LOCK-OUT ON THE DHU-11.
3666 022372 016704 157644 MOV #52525,R3 ;INITIALISE THE SKIP SELF-TEST CODE.
3667 022376 010124 4$: DEC R1 ;SELECT THE NEXT SET OF DEVICE REGISTERS.
3668 022400 010324 MOV CSRA,R4 ;GET THE ADDRESS OF THE CSR OF THE DUT.
3669 022402 020467 157652 MOV R1,(R4)+ ;SELECT A BANK OF DUT REGISTERS.
3670 022406 103774 6$: MOV R3,(R4)+ ;WRITE THE CODE TO A DUT REGISTER.
3671 022410 032701 000017 CMP R4,TXBFCA ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
3672 022414 001365 BLO 6$ ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
3673 BIT #17,R1 ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
3674 022416 004736 BNE 4$ ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.
3675 022420 000207 60$: PASS ;RESTORE GPRS.
;C,@(SP)+ ;RETURN TO PREG05 SUBRT.
RTS PC JSR

```



```

3677 .SBTTL GLOBAL SUBROUTINE - SWAPO -
3678 ;* *****
3679 ;* - SWAP GPRS WITH GPR SET 0 ROUTINE -
3680 ;* THIS SUBROUTINE SWAPS THE PRESENT CONTENTS OF GPRS R1 THRU R5 WITH
3681 ;* THE CONTENTS OF THE NUMBER ZERO GPR SAVE AREA. THE CONTENTS OF R0
3682 ;* ARE NOT ALTERED BY THIS SUBROUTINE.
3683 ;*
3684 ;* INPUTS: GPR CONTENTS R1 THRU R5.
3685 ;* GPRSOB - LABEL AT BASE OF GPR SAVE AREA NUMBER ZERO.
3686 ;*
3687 ;* OUTPUTS: R1 THRU R5 CONTAIN THE PREVIOUS CONTENTS OF GPR SAVE AREA
3688 ;* ZERO WORDS 1 THRU 5 RESPECTIVELY.
3689 ;* GPRSO - GPR SAVE AREA 0 WORDS 1 THRU 5, CONTAIN PREVIOUS
3690 ;* CONTENTS OF GPRS R1 THRU R5 RESPECTIVELY.
3691 ;*
3692 ;* CALLING SEQUENCE: JSR PC,SWAPO
3693 ;*
3694 ;* COMMENTS: THE STATE OF THE CARRY FLAG IS NOT ALTERED BY THIS ROUTINE.
3695 ;*
3696 ;* SUBORDINATE ROUTINES CALLED: NONE.
3697 ;* - *****
3698
3699 022422 010046 SWAPO:: MOV R0,-(SP) ;SAVE THE CONTENTS OF R0.
3700 ;*
3701 ;* LOAD THE STACK FROM THE GPRS.
3702 ;*
3703 022424 010146 MOV R1,-(SP) ;SAVE THE CONTENTS OF R1.
3704 022426 010246 MOV R2,-(SP) ;SAVE THE CONTENTS OF R2.
3705 022430 010346 MOV R3,-(SP) ;SAVE THE CONTENTS OF R3.
3706 022432 010446 MOV R4,-(SP) ;SAVE THE CONTENTS OF R4.
3707 022434 010546 MOV R5,-(SP) ;SAVE THE CONTENTS OF R5.
3708 ;*
3709 ;* LOAD THE GPRS FROM THE GPR SAVE AREA 0.
3710 ;*
3711 022436 012700 002450 MOV *GPRSOB,R0 ;GET THE BASE ADDRESS OF GPR SAVE AREA 0.
3712 022442 012001 MOV (R0)+,R1 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 1.
3713 022444 012002 MOV (R0)+,R2 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 2.
3714 022446 012003 MOV (R0)+,R3 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 3.
3715 022450 012004 MOV (R0)+,R4 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 4.
3716 022452 012005 MOV (R0)+,R5 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 5.
3717 ;*
3718 ;* LOAD THE GPR SAVE AREA 0 FROM THE STACK.
3719 ;*
3720 022454 012640 MOV (SP)+,(R0) ;LOAD GPR SAVE AREA 0 WORD 5 WITH SAVED R5.
3721 022456 012640 MOV (SP)+,-(R0) ;LOAD GPR SAVE AREA 0 WORD 4 WITH SAVED R4.
3722 022460 012640 MOV (SP)+,-(R0) ;LOAD GPR SAVE AREA 0 WORD 3 WITH SAVED R3.
3723 022462 012640 MOV (SP)+,-(R0) ;LOAD GPR SAVE AREA 0 WORD 2 WITH SAVED R2.
3724 022464 012640 MOV (SP)+,-(R0) ;LOAD GPR SAVE AREA 0 WORD 1 WITH SAVED R1.
3725 ;*
3726 022466 012600 MOV (SP)+,R0 ;RESTORE THE INITIAL VALUE OF R0.
3727 ;*
3728 022470 000207 RTS PC

```

```

3730 .SBTTL GLOBAL SUBROUTINE - TSABRT -
3731 ;* *****
3732 ;* - TEST ABORT ROUTINE -
3733 ;* THIS SUBROUTINE IS USED WHEN A NON-TEST RELATED ERROR HAS BEEN FOUND
3734 ;* DURING THE EXECUTION OF THE CURRENT TEST.
3735 ;* IT IS USED TO INFORM THE OPERATOR THAT THE CURRENT TEST HAS BEEN
3736 ;* ABORTED.
3737 ;*
3738 ;* INPUTS: ERRMSG - CONTAINS THE NAME OF THE CURRENT TEST.
3739 ;*          ERRNBR - CONTAINS THE CORRECT ERROR NUMBER.
3740 ;*          THE REMAINDER OF THE ERRBLK IS CORRECTLY INITIALISED.
3741 ;*
3742 ;* OUTPUTS: MESSAGES ARE REPORTED TO THE OPERATOR.
3743 ;*
3744 ;* CALLING SEQUENCE: JSP PC,TSABRT
3745 ;*
3746 ;* COMMENTS:
3747 ;*
3748 ;* SUBORDINATE ROUTINES CALLED: ER1603.
3749 ;* - - - - -
3750
3751 022472 TSABRT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3752 022472 004567 161300 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3753 022476 012701 022514 MOV #2$,R1 ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
3754 022502 012767 016526 161264 MOV #ER1603,ERRBLK ;SET-UP THE ERROR REPORTING ROUTINE.
3755 022510 104460 ;
3756 022512 000432 ; >>>> ERROR <<<<. TRAP C$E'RROR
3757 022514 040 116 117 2$: BR 60$ ;
3758 022517 116 055 122 .ASCIZ / NON-RELATED TEST ERROR FOUND DURING TEST EXECUTION/
3759 022522 105 114 101
3760 022525 124 105 104
3761 022530 040 124 105
3762 022533 123 124 040
3763 022536 105 122 122
3764 022541 117 122 040
3765 022544 106 117 125
3766 022547 116 104 040
3767 022552 104 125 122
3768 022555 111 116 107
3769 022560 040 124 105
3770 022563 123 124 040
3771 022566 105 130 105
3772 022571 103 125 124
3773 022574 111 117 116
3774 022577 000
3775 .EVEN
3776 60$: PASS
3777 RTS PC JSR ;RESTORE GPRS.
3778 022600 004736 PC,0(SP). ;RETURN TO PREG05 SUBRT.
3779 022602 000207

```

```

3761 .SBTTL GLOBAL SUBROUTINE - TXDSBL -
3762 ;* *****
3763 ;* - TRANSMITTER DISABLE -
3764 ;* THIS SUBROUTINE IS USED TO DISABLE TRANSMISSION ON SELECTED LINES BY,
3765 ;* CLEARING THE ASSOCIATED TX.ENABLE BIT ON THE OUT.
3766 ;*
3767 ;* INPUTS: R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO CLEAR TX.ENABLE.
3768 ;* CSRA - CONTAINS THE ADDRESS OF THE OUT CSR.
3769 ;* IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
3770 ;* NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVATLABLE.
3771 ;* TXAD2A - CONTAINS THE ADDRESS OF THE TBUFFAD2 REGISTER.
3772 ;*
3773 ;* OUTPUTS: R5 - BIT'S SET INDICATE THE INITIAL STATES OF ALL TX.ENABLE BITS.
3774 ;* TBUFFAD2 - THE STATE OF THE TX.ENABLE BIT MAY BE ALTERED.
3775 ;* THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
3776 ;*
3777 ;* CALLING SEQUENCE: JSR PC,TXDSBL
3778 ;*
3779 ;* COMMENTS:
3780 ;*
3781 ;* SUBORDINATE ROUTINES CALLED: NONE.
3782 ;* *****
3783
3784 TXDSBL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3785 022604 004567 161166 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3786 022610 010500 MOV R5,R0 ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
3787 022612 012701 000001 MOV #BIT0,R1 ;INITIALIZE THE SELECTED LINE BIT MASK.
3788 022616 016702 157434 MOV TXAD2A,R2 ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
3789 022622 005202 INC R2 ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
3790 022624 012703 000020 MOV #NUMLNS,R3 ;GET MAXIMUM LINE NUMBER PLUS ONE.
3791 022630 016704 157454 MOV IESTAT,R4 ;GET THE STATES OF THE INT ENABLE BITS.
3792 022634 005005 CLR R5 ;LOG POSSIBLE TX DISABLED ON ALL LINES.
3793
3794 ; SELECT EVERY LINE IN TURN, AND LOG THE STATE OF EACH TX.ENABLE BIT.
3795 022636 010477 157400 2$: MOV R4,#CSRA ;WRITE TO OUT CSR TO SELECT LINE REGISTERS.
3796 022642 105712 TSTB (R2) ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
3797 022644 100001 BPL 4$ ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
3798 022646 050105 BIS R1,R5 ;LOG TX ENABLE BIT SET FOR SELECTED LINE.
3799
3800 ; CLEAR TX.ENABLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX DISABLE
3801 ; LINE BIT MAP.
3802
3803 022650 030100 4$: BIT R1,R0 ;CHECK STATE OF DISABLE LINE BIT MAP.
3804 022652 001402 BEQ 6$ ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
3805 022654 142712 000200 BICB #BIT7,(R2) ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
3806 022660 005204 INC R4 ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
3807 022662 006301 ASL R1 ;SHIFT BIT MAP FOR NEXT LINE.
3808 022664 005303 DEC R3 ;DECREMENT LINE NUMBER.
3809 022666 001363 BNE 2$ ;LOOP TO CHECK NEXT LINE.
3810
3811 022670 60$: PASS R5 ;RESTORE GPRS,EXCEPT
3812 022670 010566 000014 MOV R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
3813 022674 004736 JSR PC,(SP) ;RETURN TO PREG05 SUBRT.
3813 022676 000207 RTS PC ;R5 PREVIOUS STATES OF ALL TX.ENABLE BITS.

```

```

3815 .SBTTL GLOBAL SUBROUTINE - TXENBL -
3816 ;* *****
3817 ;* - TRANSMITTER ENABLE -
3818 ;* THIS SUBROUTINE IS USED TO ENABLE TRANSMISSION ON SELECTED LINES BY
3819 ;* SETTING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
3820 ;*
3821 ;* INPUTS: R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO SET TX.ENABLE.
3822 ;* CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
3823 ;* IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
3824 ;* NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
3825 ;* TXAD2A - CONTAINS THE ADDRESS OF THE TBUFFAD2 REGISTER.
3826 ;*
3827 ;* OUTPUTS: R5 - BIT'S SET INDICATE PREVIOUSLY DISABLED LINES.
3828 ;* TBUFFAD2 - THE STATE OF THE TX.ENABLE BIT MAY BE ALTERED.
3829 ;* THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
3830 ;*
3831 ;* CALLING SEQUENCE: JSR PC,TXENBL
3832 ;*
3833 ;* COMMENTS:
3834 ;*
3835 ;* SUBORDINATE ROUTINES CALLED: NONE.
3836 ;* - *****
3837
3838 TXENBL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3839 022700 004567 161072 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3840 022704 010500 MOV R5,R0 ;COPY BIT MAP OF LINES TO ENABLE.
3841 022706 012701 000001 MOV #BIT0,R1 ;INITIALIZE THE SELECTED LINE BIT MASK.
3842 022712 016702 15734C MOV TXAD2A,R2 ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
3843 022716 005202 INC R2 ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
3844 022720 012703 000020 MOV #NUMLNS,R3 ;GET MAXIMUM LINE NUMBER.
3845 022724 016704 157360 MOV IESTAT,R4 ;GET THE STATES OF THE INT ENABLE BITS.
3846 022730 005005 CLR R5 ;CLEAR TX.ENABLE BIT LOG OF DISABLED LINES.
3847
3848 ;*
3849 ;* SELECT EVERY LINE IN TURN,AND LOG ANY TX.ENABLE BIT THAT IS CLEAR.
3850 022732 010477 157304 2$: MOV R4,&CSRA ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
3851 022736 105712 TSTB (R2) ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
3852 022740 100401 BMI 4$ ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
3853 022742 050105 BIS R1,R5 ;LOG TX ENABLE BIT CLEAR FOR SELECTED LINE.
3854
3855 ;*
3856 ;* SET TX.ENABLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX ENABLE
3857 ;* LINE BIT MAP.
3858 022744 030100 4$: BIT R1,R0 ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
3859 022746 001402 BEQ 6$ ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
3860 022750 152712 000200 BISH #BIT7,(R2) ;ENABLE TRANSMISSION ON SELECTED LINE.
3861 022754 005204 6$: INC R4 ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
3862 022756 006301 ASL R1 ;SHIFT BIT MAP FOR NEXT LINE.
3863 022760 005303 DEC R3 ;DECREMENT LINE NUMBER.
3864 022762 001353 BNE 2$ ;LOOP TO CHECK NEXT LINE.
3865 022764 60$: PASS R5 ;RESTORE GPRS,EXCEPT
3866 022764 010566 000014 MOV R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
3867 022770 004736 JSR PC,&(SP) ;RETURN TO PREG05 SUBRT.
3868 022772 000207 RTS PC ;R5 - LINE BIT MAP CORRESPONDING TO THE
; PREVIOUS LINES THAT WERE DISABLED.

```

```

3870 ,SBTTL GLOBAL SUBROUTINE - TXIEO -
3871 ;* *****
3872 ;* - TRANSMITTER INTERRUPT DISABLE -
3873 ;* THIS ROUTINE IS USED TO DISABLE TRANSMITTER INTERRUPTS IN THE DHU11.
3874 ;*
3875 ;* INPUTS: NONE.
3876 ;*
3877 ;* OUTPUTS: THE TX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
3878 ;* IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3879 ;* ENABLE BITS.
3880 ;*
3881 ;* CALLING SEQUENCE: JSR PC,TXIEO
3882 ;*
3883 ;* COMMENTS: THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3884 ;* THE DUT CSR ARE DESTROYED.
3885 ;*
3886 ;* SUBORDINATE ROUTINES CALLED: NONE.
3887 ;* *****
3888 TXIEO:: MOV RO,-(SP) ;SAVE CONTENTS OF RO ON THE STACK.
3889 GETPRI -(SP) ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
3890 ; TRAP C$GPRI
3891 ; MOV RO,-(SP)
3892 ; SETPRI #PRI07 ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
3893 ; MOV #PRI07,RO
3894 ; TRAP C$SPRI
3895 ; BIC #177677,IESTAT ;CLEAR TX.INT.ENBL BIT IN IESTAT.
3896 ; MOV IESTAT,@CSRA ;DISABLE TX INTERRUPTS.
3897 ; SETPRI (SP)+ ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
3898 ; MOV (SP)+,RO
3899 ; TRAP C$SPRI
3900 ; MOV (SP)+,RO
3901 ; TRAP C$SPRI
3902 ;
3903 ; RESTORE RO.
3904 ; MOV (SP)+,RO
3905 ; TRAP C$SPRI
3906 ;
3907 ; RTS PC

```

```

3897      .SBTTL GLOBAL SUBROUTINE                      - TXIE1 -
3898      ;** *****
3899      ;*          - TRANSMITTER INTERRUPT ENABLE -
3900      ;*          THIS ROUTINE IS USED TO ENABLE TRANSMITTER INTERRUPTS IN THE DHU11.
3901      ;*
3902      ;* INPUTS:      NONE.
3903      ;*
3904      ;* OUTPUTS:     THE TX INT.ENBL BIT IS SET IN THE DUT CSR.
3905      ;*               IESTAT -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3906      ;*               ENABLE BITS.
3907      ;*
3908      ;* CALLING SEQUENCE:  JSR      PC,TXIE1
3909      ;*
3910      ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3911      ;*               THE DUT CSR ARE DESTROYED.
3912      ;*
3913      ;* SUBORDINATE ROUTINES CALLED: NONE.
3914      ;** *****
3915
3916 023034 052767 040000 157246 TXIE1:: BIS    @BIT14,IESTAT ;SET TX.INT.ENBL BIT IN IESTAT.
3917 023042 042767 137677 157240      BIC    @137677,IESTAT ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
3918 023050 016777 157234 157164      MOV    IESTAT,@CSRA  ;ENABLE TX INTERRUPTS.
3919 023056 000207          RTS     PC

```



```

3921 .SBTTL GLOBAL SUBROUTINE - UNSDIV -
3922 ;+ *****
3923 ;+ - UNSIGNED DIVIDE ROUTINE -
3924 ;+ THIS SUBROUTINE IS USED TO DIVIDE A 32 BIT UNSIGNED DIVIDEND BY A
3925 ;+ 16 BIT UNSIGNED DIVISOR GIVING A 16 BIT QUOTIENT. ALL NUMBERS ARE
3926 ;+ CONSIDERED TO BE UNSIGNED. A SUCCESS FLAG IS NOT SET ON RETURN IF
3927 ;+ THE QUOTIENT WAS TOO BIG TO BE CONTAINED IN 16 BITS.
3928 ;+
3929 ;+ INPUTS: R1 - THE DIVISOR, UNSIGNED, 16 BITS.
3930 ;+ R2 - MOST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
3931 ;+ R3 - LEAST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
3932 ;+
3933 ;+ OUTPUTS: R1 - QUOTIENT, UNSIGNED, 16 BITS (177777 IF OVERFLOW).
3934 ;+ CARRY - SUCCESS FLAG, SET IF COMPLETE QUOTIENT FITS IN 16 BITS.
3935 ;+
3936 ;+ CALLING SEQUENCE: JSR PC,UNSDIV
3937 ;+
3938 ;+ COMMENTS: IF THE DIVISOR IS 0 THE QUOTIENT IS RETURNED AS ALL ONES
3939 ;+ (177777) AND THE CARRY IS CLEAR REGARDLESS OF THE DIVIDEND.
3940 ;+
3941 ;+ SUBORDINATE ROUTINES CALLED: NONE.
3942 ;+
3943 ;+ *****
3944 UNSDIV:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3945 ; JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3946 ;
3947 ; CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
3948 ;+
3948 023064 010204 MOV R2,R4 ;GET MSW OF DIVIDEND FOR SUBTRACT.
3949 023066 160104 SUB R1,R4 ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
3950 023070 103403 BCS 2$ ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
3951 023072 012701 177777 MOV 0-1,R1 ;SET QUOTIENT TO ALL ONES (177777).
3952 023076 000442 BR 60$ ;EXIT WITH CARRY CLEAR.
3953 ;+
3954 ; SET UP COUNTERS AND VARIOUS WORKING GPRS.
3955 ;+
3956 023100 005004 2$: CLR R4 ;CLEAR THE LSW OF THE DIVISOR.
3957 023102 000241 CLC ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
3958 023104 006001 ROR R1 ;DIVISOR BY
3959 023106 006004 ROR R4 ;2(UNSIGNED)
3960 023110 012700 000020 MOV #16,,R0 ;SET UP INITIAL SHIFT COUNT TO 16.
3961 ;+
3962 ; THE SUBTRACT AND SHIFT LOOP.
3963 ;+
3964 023114 010246 4$: MOV R2,-(SP) ;SAVE MSWORD OF DIVIDEND.
3965 023116 010346 MOV R3,-(SP) ;SAVE LSWORD OF DIVIDEND.
3966 023120 160403 SUB R4,R3 ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
3967 023122 005602 SBC R2 ;MSWORD DIVIDEND - BORROW.
3968 023124 103402 BCS 6$ ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
3969 023126 160102 SUB R1,R2 ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
3970 023130 103003 BCC 8$ ;IF NO BORROW, IT WENT, CARRY IS CLEAR.
3971 ;+
3972 ; IT DIDN'T GO, SO WE SHIFT A 1 INTO THE QUOTIENT (COMPLEMENTED LATER).
3973 ; CARRY IS SET.
3974 ;+
3975 023132 012603 6$: MOV (SP)+,R3 ;RESTORE LSWORD OF DIVIDEND.
3976 023134 012602 MOV (SP)+,R2 ;RESTORE MSWORD OF DIVIDEND.

```

N8

CZDHUBO DHU-11 FUNC 1ST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 80-1
GLOBAL SUBROUTINE UNSDIV

SEG 104

```

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

```



```

4017 .SBTTL GLOBAL SUBROUTINE - WAIBIC -
4018 .....
4019 * - WAIT FOR BIT CLEAR ROUTINE -
4020 * THIS SUBROUTINE WAITS FOR THE SPECIFIED BIT TO BECOME CLEAR. IF THE
4021 * SPECIFIED BIT GOES TO A CLEAR STATE WITHIN THE SPECIFIED TIME-OUT
4022 * PERIOD A SUCCESS INDICATION IS RETURNED BY THIS ROUTINE.
4023 * THE LAST VALUE WHICH IS READ LOOKING FOR THE CONDITION IS RETURNED TO
4024 * ALLOW THE USE OF THIS ROUTINE TO LOOK FOR DESTRUCTIVE READ CONDITIONS.
4025 *
4026 * INPUTS: R1 - TIME-OUT VALUE AND BIT NUMBER INDICATION:
4027 *          BITS 15 THRU 12 - NUMBER OF BIT TO TEST (RANGE 0 THRU 15).
4028 *          BITS 11 THRU 0 - TIME-OUT VALUE IN MILLI-SECONDS (4095 MAX).
4029 *          R2 - ADDRESS OF WORD CONTAINING THE BIT TO TEST.
4030 *          MSLCNT.
4031 *
4032 * OUTPUTS: R2 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
4033 *          CARRY - SUCCESS FLAG (CARRY SET IF BIT CLR BEFORE TIME-OUT).
4034 *
4035 * CALLING SEQUENCE: MOV 0130040,R1 ;PASS BIT 11 (13 OCTAL) AND
4036 *                   MOV 01 LABEL,R2 ; 32 (40 OCTAL) MS DELAY.
4037 *                   JSR PC,WAIBIC ;TEST BIT IN WORD AT "LABEL".
4038 *                   ;WAIT 32 MS FOR BIT 11 TO CLR.
4039 *
4040 * COMMENTS:
4041 *
4042 * SUBORDINATE ROUTINES CALLED: MSLGET.
4043 .....
4044
4045 023214 WAIBIC:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
4046 023214 004567 160556 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4047 023220 010204 MOV R2,R4 ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
4048 023222 010102 MOV R1,R2
4049 023224 042701 170000 BIC 0170000,R1 ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
4050 023230 042702 007777 BIC 07777,R2 ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
4051 023236 000302 SWAB R2 ;PUT LINE NUMBER FIELD IN LSHYTE.
4052 023240 006202 ASR R2 ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
4053 023242 006202 ASR R2 ; POSITION TO USE IT AS A WORD TABLE OFFSET.
4054 023244 016202 002410 MOV BITTAB(R2),R2 ; FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
4055 023250 005003 CLR R3 ;GET BIT MAP OF LINE TO TEST FROM TABLE.
4056 023252 004767 174706 JSR PC,MSLGET ;INDICATE THAT THE BIT SHOULD BE CLR.
4057 ;WAIT FOR THE BIT TO BE CLR WITHIN TIME OUT.
4058 023256 010002 MOV R0,R2 ; CARRY IS CORRECT UPON MSLGET RETURN.
4059 023260 6031 PASS R2 ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
4060 023260 010266 000006 MOV R2,R2SLOT(SP) ;RESTORE GPRS, EXCEPT THE FOLLOWING:
4061 023266 000207 JSR PC,0(SP) ;PUT R2 IN STACK SLOT.
4061 023266 000207 RTS PC ;RETURN TO PREG05 SUBRT.
4061 023266 000207 ; R2 - LAST VALUE READ LOOKING FOR CONDITION.
4061 023266 000207 ; CARRY - SUCCESS FLAG (SET IF BIT FOUND CLR).

```

```

4063 .SRTTL GLOBAL SUBROUTINE - WAIBIS
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 023270
4092 023270 004567 160502
4093 023274 010204
4094 023300 042701 170000
4095 023304 042701 007777
4096 023310 000301
4097 023312 006202
4098 023314 006202
4099 023316 006202
4100 023320 016202 002410
4101 023324 010203
4102 023326 004757 174632
4103
4104 023332 010002
4105 023334
4106 023334 010266 000006
4107 023340 004736
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

```

```

4109 .SRTTL GLOBAL SUBROUTINE - WDPDR -
4110 *****
4111 - WRITE DATA PATTERN TO DEVICE REGISTERS -
4112 THIS ROUTINE WRITES A ROTATED DATA PATTERN TO EACH OF THE 6 DEVICE
4113 REGISTERS OF EACH ACTIVE LINE OF THE DEVICE UNDER TEST.
4114 THE DATA PATTERN IS ROTATED ONCE AFTER EACH WRITE TO A DEVICE REGISTER
4115 ON A PARTICULAR LINE. THE STARTING DATA PATTERN FOR EACH LINE
4116 IS ROTATED ONCE AFTER WRITING ALL THE REGISTERS ON A PARTICULAR
4117 LINE. THIS LEADS TO THE FOLLOWING DATA PATTERN:
4118 LINE 0, REGISTER 0 - SHIFTED 0 BIT POSITIONS
4119 LINE 0, REGISTER 1 - SHIFTED 1 BIT POSITION
4120
4121 LINE 1, REGISTER 0 - SHIFTED 1 BIT POSITION
4122 LINE 2, REGISTER 1 - SHIFTED 2 BIT POSITIONS
4123
4124 ANY BITS FIELDS IN THE DEVICE REGISTERS THAT CANNOT BE ALTERED
4125 ARE MASKED OUT OF THE DATA PATTERN BEFORE IT IS WRITTEN.
4126 THIS ROUTINE WILL USE EITHER MOV, MOVB, BIS, BISB, BIC, OR BICB
4127 INSTRUCTIONS. THE UPPER OR LOWER BYTE CAN BE SPECIFIED FOR WRITING.
4128
4129 INPUTS: R2 - USED TO PASS IN THE DATA PATTERN TO BE ROTATED & WRITTEN.
4130 R3 - BYTE INDICATOR (- >> LO BYTE, + >> HI BYTE, 0 >> BOTH).
4131 R4 - OPERATION TYPE INDICATOR ( >> BIC, + >> BIS, 0 >> MOV).
4132 ACTLNS - BIT MAP OF THE ACTIVE LINES ON THE DEVICE UNDER TEST.
4133 CSRA - CONTAINS THE CSR ADDRESS OF THE DEVICE UNDER TEST.
4134 DRADRT - BASE ADDRESS OF DEVICE REGISTER ADDRESS TABLE.
4135 LPRO - EQUATED TO LPR REG OFFSET FROM DEVICE CSR ADDRESS.
4136 NUMLNS - NUMBER OF LINES ON THE DEVICE UNDER TEST.
4137 TXBFCO - EQUATED TO TBUFFT REG OFFSET FROM DEVICE CSR ADDRESS.
4138 UNBITB - BASE ADDRESS OF THE UNUSED BIT TABLE.
4139
4140 OUTPUTS: DEVICE REGISTERS ON ALL ACTIVE DEVICE LINES ARE MODIFIED.
4141
4142 CALLING SEQUENCE: JSR PC, WDPDR
4143
4144 COMMENTS: THIS ROUTINE DOES NOT WRITE DATA TO THE FOLLOWING REGISTERS,
4145 RBUF
4146 RXTIMER
4147 STAT
4148 FIFOSIZE
4149 FIFODATA
4150 THE CSR IS CLEARED EXCEPT FOR THE IND.ADR.REG FIELD.
4151
4152 SUBORDINATE ROUTINES CALLED: ROLDAP.
4153 *****
4154
4155 023344 WDPDR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
023344 004567 160426 JSR R5, PREG05 ;CALL REGISTER SAVE SUBRT.
4156
4157 ; SET UP OUTER LOOP WHICH WRITES THE DATA PATTERN TO EACH LINE'S REGISTERS
4158
4159 023350 005005 CLR R5 ;CLEAR LINE COUNTER TO SELECT LINE 0.
4160
4161 ; THE OUTER LOOP FOLLOWS. EACH PASS THROUGH THIS LOOP WRITES DATA TO ALL OF
4162 ; THE DEVICE REGISTERS FOR A PARTICULAR LINE IF IT IS ACTIVE.
4163
4164 023352 010204 ; MOV R2, R4 ;SAVE THE OUTER LOOP DATA PATTERN.

```

```

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

```

4222	023536	005205		INC	R5		;COUNT THIS LINE.
4223	023540	020527	000020	CMP	R5,#NUMLNS		;COMPARE LINE COUNT WITH NUMBER OF LINES.
4224	023544	002702		BLT	2\$		;LOOP IF SOME LINES NOT DONE.
4225							
4226	023546		60\$:	PASS			;RESTORE GPRS.
	023546	004736				JSR	PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
4227	023550	000207		RTS	PC		

4229 .SBTTL GLOBAL SUBROUTINE - WTWLNC -
4230 ;* *****
4231 ;* - LINE CONTROL REGISTER SETUP ROUTINE -
4232 ;* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
4233 ;* CONTROL REGISTERS (LNCTRL) TO THE SPECIFIED STATE. ONLY THE LNCTRLS
4234 ;* FOR THE SPECIFIED LINES ARE ALTERED.
4235 ;*
4236 ;* INPUTS: R0 - NEW LINE PARAMETERS.
4237 ;* R5 - BIT MAP OF LINES TO BE ALTERED.
4238 ;* CSRA - CONTAINS ADDRESS OF THE DUT CSR.
4239 ;* IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
4240 ;* ENABLE BITS IN THE CSR.
4241 ;* LNCTRA - CONTAINS ADDRESS OF THE DUT LNCTRL REGISTERS.
4242 ;*
4243 ;* OUTPUTS: LNCTRL - SPECIFIED DUT LINE CONTROL REGISTERS ARE ALTERED.
4244 ;*
4245 ;* CALLING SEQUENCE: JSR PC,WTWLNC
4246 ;*
4247 ;* COMMENTS:
4248 ;*
4249 ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
4250 ;* - *****
4251
4252 023552 WTWLNC:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
023552 004567 160220 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4253 ;*
4254 ;* SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
4255 ;*
4256 023556 016701 156470 MOV LNCTRA,R1 ;SET UP THE REGISTER ADDRESS PARAMETER.
4257 023562 010002 MOV R0,R2 ;SET UP THE DESIRED REGISTER CONTENTS.
4258 023564 010503 MOV R5,R3 ;SET UP THE BIT MAP OF LINES TO ALTER.
4259 023566 012704 177777 MOV 0-1,R4 ;SELECT ALL REGISTER BITS TO BE ALTERED.
4260 ;*
4261 ;* CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
4262 ;*
4263 023572 004767 173656 JSR PC,ALTFLD ;ALTER THE REGISTER CONTENTS.
4264
4265 023576 60\$: PASS ;RESTORE GPRS.
023576 004736 JSR PC,@(SP), ;RETURN TO PREG05 SUBRT.
4266 023600 000207 RTS PC


```

4268      ;SBTTL GLOBAL SUBROUTINE - WTWLPR -
4269      ;* *****
4270      ;* - LINE PARAMETER REGISTER SETUP ROUTINE -
4271      ;* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
4272      ;* PARAMETER REGISTERS (LPR) TO THE SPECIFIED STATE. ONLY THE LPRS FOR
4273      ;* THE SPECIFIED LINES ARE ALTERED.
4274      ;*
4275      ;* INPUTS:      RO - NEW LINE PARAMETERS.
4276      ;*              R5 - BIT MAP OF LINES TO BE ALTERED.
4277      ;*              CSRA - CONTAINS ADDRESS OF THE DUT CSR.
4278      ;*              IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
4279      ;*              ENABLE BITS IN THE CSR.
4280      ;*              LPRA - CONTAINS ADDRESS OF THE DUT LPR.
4281      ;*
4282      ;* OUTPUTS:      LPR - SPECIFIED DUT LINE PARAMTER REGISTERS ARE ALTERED.
4283      ;*
4284      ;* CALLING SEQUENCE:  JSR      PC,WTWLPR
4285      ;*
4286      ;* COMMENTS:
4287      ;*
4288      ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
4289      ;* - *****
4290
4291      023602      WTWLPR:: SAVE          ;SAVE CONTENTS OF GPRS R0 THRU R5.
4292      023602      004567      160170      JSR      R5,PREG05      ;CALL REGISTER SAVE SUBRT.
4293
4294      ;* SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
4295      ;*
4296      023606      016701      156434      MOV      LPRA,R1      ;SET UP THE REGISTER ADDRESS PARAMETER.
4297      023612      010002      MOV      RO,R2      ;SET UP THE DESIRED REGISTER CONTENTS.
4298      023614      010503      MOV      R5,R3      ;SET UP THE BIT MAP OF LINES TO ALTER.
4299      023616      012704      177777      MOV      0-1,R4      ;SELECT ALL REGISTER BITS TO BE ALTERED.
4300
4301      ;* CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
4302      023622      004767      173626      JSR      PC,ALTFLD      ;ALTER THE REGISTER CONTENTS.
4303
4304      023626      004736      60$:      PASS          ;RESTORE GPRS.
4305      023630      000207      RTS      PC      JSR      PC,@(SP) ;RETURN TO PREG05 SUBRT.

```

```

4307 .SBTTL INTERRUPT SERVICE ROUTINE - CACHRX -
4308 ;* *****
4309 ;* CATCH RECEIVER INTERRUPT.
4310 ;* THIS ROUTINE IS USED IN SEVERAL TESTS, TO LOG A COUNT OF THE
4311 ;* NUMBER OF RECEIVER INTERRUPTS THAT OCCUR.
4312 ;*
4313 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR.
4314 ;* RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERRUPTS
4315 ;* THAT OCCURRED.
4316 ;*
4317 ;* OUTPUTS: RXINTC - CONTAINS THE UPDATED INTERRUPT COUNT.
4318 ;*
4319 ;* CALLING SEQUENCE: PUT THE ADDRESS OF THE LABEL CACHRX IN THE VECTOR
4320 ;* LOCATION.
4321 ;*
4322 ;* COMMENTS:
4323 ;*
4324 ;* SUBORDINATE ROUTINES CALLED: NONE
4325 ;* *****
4326
4327 CACHRX::SAVE
4328 023632 004567 160140 ;SAVE CONTENTS OF GPRS R0 THRU R5.
023632 016701 156452 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4329 023636 016701 156452 MOV RXINTC,R1 ;GET THE RECEIVER INTERRUPT COUNT
4330 023642 005201 INC R1 ;INCREMENT THE COUNT
4331 023644 102001 BVC 2$ ;BRANCH IF NO OVERFLOW OCCURRED
4332 023646 005301 DEC R1 ;RESET THE COUNT TO 177777
4333 023650 010167 156440 2$: MOV R1,RXINTC ;SAVE NEW COUNT VALUE
4334 023654 004736 60$: PASS ;RESTORE GPRS.
023654 004736 JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
4335 023656 000002 RTI

```



```

4337 .SBTTL  INTERRUPT SERVICE ROUTINE      - CACHTX -
4338 ;* *****
4339 ;* - CATCH TRANSMITER INTERRUPT.
4340 ;* THIS ROUTINE IS USED IN SEVERAL TESTS, TO LOG A COUNT OF THE
4341 ;* NUMBER OF TRANSMISSION INTERRUPTS THAT OCCUR.
4342 ;*
4343 ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR.
4344 ;*              TXINTC - HOLDS THE COUNT OF THE NUMBER OF TX INTERRUPTS
4345 ;*                  THAT OCCURRED.
4346 ;*
4347 ;* OUTPUTS:     TXINTC - CONTAINS THE UPDATED INTERRUPT COUNT.
4348 ;*
4349 ;* CALLING SEQUENCE:  PUT THE ADDRESS OF THE LABEL CACHTX IN THE VECTOR
4350 ;*                  LOCATION.
4351 ;*
4352 ;* COMMENTS:
4353 ;*
4354 ;* SUBORDINATE ROUTINES CALLED: NONE
4355 ;* *****
4356 ;*
4357
4358 023660 CACHTX::SAVE                      ;SAVE CONTENTS OF GPRS R0 THRU R5.
      023660 004567 160112                R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4359 023664 016701 156436                MOV     TXINTC,R1 ;GET THE TRANSMISSION INTERRUPT COUNT
      023670 005201                        INC     R1      ;INCREMENT THE COUNT
4361 023672 102001                        BVC     2$        ;BRANCH IF NO OVERFLOW OCCURRED
4362 023674 005301                        DEC     R1        ;RESET THE COUNT TO 177777
4363 023676 010167 156424                2$: MOV     R1,TXINTC ;SAVE NEW COUNT VALUE
4364 023702                                60$: PASS      ;RESTORE GPRS.
      023702 004736                        JSR     PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
4365 023704 000002                        RTI
  
```

```

4367 .SBTTL INTERRUPT SERVICE ROUTINE - CLKINT -
4368 ; *****
4369 ; THIS ROUTINE IS EXECUTED CLKHRZ TIMES PER SECOND. IT DECREASES THE
4370 ; TWO TIMER COUNTERS DOWN TO ZERO.
4371 ;
4372 ; INPUTS: TIMER1 - TIMER COUNTER #1.
4373 ;         TIMER2 - TIMER COUNTER #2.
4374 ;         TIMER3 - TIMER COUNTER FOR CALL OF BREAK MACRO.
4375 ;
4376 ; OUTPUTS: THE 2 TIMER COUNTERS ARE DECREMENTED IF THEY ARE NOT ZERO.
4377 ;
4378 ; CALLING SEQUENCE: PUT #CLKINT IN THE CLOCK INTERRUPT VECTOR SLOT.
4379 ;                  PUT THE DESIRED TIME PERIOD (SECONDS TIMES CLKHRZ) IN
4380 ;                  EITHER TIMER1 OR TIMER2 AND POLL THE RESPECTIVE TIMER
4381 ;                  COUNTER TO DETECT ITS GOING TO 0 ON TIME-OUT.
4382 ;
4383 ; COMMENTS: THE 2 COUNTERS WILL NOT WRAPAROUND BUT WILL STOP AT 0. THIS
4384 ;            ALLOWS THE DETECTION OF A TIME-OUT ANY TIME AFTER THE TIME-OUT
4385 ;            HAS OCCURRED UNTIL THE TIMER COUNTER IS SET TO ANOTHER VALUE.
4386 ;
4387 ; SUBORDINATE ROUTINES CALLED: NONE.
4388 ; - - - - -
4389
4390 023706 005767 156432 CLKINT:: TST    TIMER1      ;CHECK FOR TIMER1 AT ZERO.
4391 023712 001402      BEQ     2$      ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
4392 023714 005367 156424      DEC     TIMER1    ;DECREMENT TIME COUNT.
4393 023720 005767 156422 2$: TST    TIMER2    ;CHECK FOR TIMER2 AT ZERO.
4394 023724 001402      BEQ     4$      ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
4395 023726 005367 156414      DEC     TIMER2    ;DECREMENT TIME COUNT.
4396 023732 005367 156412 4$: DEC     TIMER3    ;DECREMENT THE BREAK COUNT.
4397 023736 001006      BNE     60$      ;EXIT IF NOT TIME TO CALL BREAK.
4398 023740 016767 156406 156402 MOV    RCOUNT,TIMER3 ;SET UP TIME TILL NEXT BREAK.
4399 023746 010046      MOV    R0,-(SP)    ;SAVE CONTENTS OF R0 FROM BREAK MACRO.
4400 023750      BREAK      ;CHECK FOR OPERATOR CONTROL/C.
4401 023752 012600      TRAP    C$BRK
4402 023754 000002      MOV    (SP)+,R0    ;RESTORE CONTENTS OF R0.
4402 023754 000002 60$: RTI

```

```

4404 .SBTTL  INTERUPT SERVICE ROUTINE          - RXBRRT -
4405 ;* *****
4406 ;* - BR LEVEL TEST RECEIVE INTERRUPT SERVICE ROUTINE -
4407 ;* THIS SERVICE ROUTINE HANDLES RECEIVE INTERRUPTS DURING THE INTERRUPT
4408 ;* BR LEVEL TEST. THIS ROUTINE COUNTS THE INTERRUPT AND SETS A FLAG
4409 ;* TO INDICATE THAT THE INTERRUPT HAS OCCURRED. IT ALSO CHECKS THE
4410 ;* FLAG WHICH INDICATES THAT A TX INTERRUPT HAS OCCURRED. IF THE TX
4411 ;* INTERRUPT FLAG IS SET, THIS ROUTINE SETS AN INTERRUPT ORDER ERROR
4412 ;* FLAG INDICATING THAT A TRANSMIT INTERRUPT WAS SERVICED BEFORE A
4413 ;* SIMULTANEOUS RECEIVE INTERRUPT.
4414 ;*
4415 ;* INPUTS:      RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERUPTS.
4416 ;*              RXINTF - RX INTERRUPT FLAGS.
4417 ;*
4418 ;* OUTPUTS:     RXINTC - CONTAINS THE UPDATED INTERUPT COUNT.
4419 ;*              RXINTF - RX INT FLAGS:
4420 ;*                  (BIT 0 SET, BIT 14 SET IF TXINTF BIT 0 IS SET.)
4421 ;*
4422 ;* CALLING SEQUENCE: PUT THE ADDRESS OF THE LABEL RXBRRT IN THE VECTOR
4423 ;* LOCATION.
4424 ;*
4425 ;* COMMENTS:    NOTE: THE FIFO IS NOT PURGED BY THIS ROUTINE.
4426 ;*
4427 ;* SUBORDINATE ROUTINES CALLED: NONE.
4428 ;* *****
4429 ;*
4430 023756 RXBRRT:: SAVE                      ;SAVE CONTENTS OF GPRS R0 THRU R5.
      023756 004567 160014                R5,PREG05 ;CALL REGISTER SAVE SUBRT.
      023756 017700 156256                MOV  @RBUFA,R0 ;READ THE CHAR OUT OF THE FIFO.
4431 023762 017700 156256                MOV  RXINTC,R1 ;GET THE INTERRUPT COUNT.
4432 023766 016701 156322                INC   R1      ;INCREMENT THE COUNT.
4433 023772 005201                    BEQ   2$        ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
4434 023774 001402                    MOV   R1,RXINTC ;SAVE NEW COUNT VALUE.
4435 023776 010167 156312                2$: MOV   RXINTF,R1 ;GET THE RX INTERRUPT FLAGS.
4436 024002 016701 156310                BIS   @BIT0,R1 ;SET THE RX INTERRUPT HAS OCCURRED FLAG.
4437 024006 052701 000001                BIT   @BIT0,TXINTF ;TEST THE "TX INT HAS OCCURRED" FLAG.
4438 024012 032767 000001 156310        BEQ   4$      ;SKIP SETTING ERROR FLAG IF NO TX INT.
4439 024020 001402                    BIS   @BIT14,R1 ;SET THE INTERRUPT ORDER ERROR FLAG.
4440 024022 052701 040000                4$: MOV   R1,RXINTF ;UPDATE THE RX INTERRUPT FLAGS.
4441 024026 010167 156264                60$: PASS  ;RESTORE GPRS.
4442 024032 004736                    JSR   PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
      024032 000002                    RTI
4443 024034

```

```

4445      .SBTTL  INTERUPT SERVICE ROUTINE      - RXINPT -
4446      ; * *****
4447      ; * - RECEIVE CHARACTER INPUT INTERRUPT SERVICE ROUTINE -
4448      ; * THIS SERVICE ROUTINE INPUTS A CHARACTER FROM THE DUT AND LOADS THE
4449      ; * CHAR (COMPLETE WITH STATUS FLAGS) INTO A RECEIVE CHAR BUFFER IN
4450      ; * MEMORY. THE INTERRUPT IS ALSO COUNTED. THE RECEIVE CHAR BUFFER IS
4451      ; * MONITORED TO ENSURE THAT IT DOES NOT OVERFLOW.
4452      ; *
4453      ; * INPUTS:      BUFEND - LABELS THE END OF THE HOST MEMORY BUFFER.
4454      ; *              BU*PTR - CONTAINS ADDRESS OF NEXT FREE BUFFER LOCATION.
4455      ; *              CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
4456      ; *              RBUFA - CONTAINS THE ADDRESS OF THE RBUF DUT REGISTER.
4457      ; *              RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERRUPTS.
4458      ; *              RXINTF - RX INTERRUPT FLAGS.
4459      ; *
4460      ; * OUTPUTS:     BU*PTR - CONTAINS UPDATED ADDRESS OF NEXT FREE BUFFER LOCATION.
4461      ; *              RXINTC - CONTAINS THE UPDATED INTERRUPT COUNT.
4462      ; *              RXINTF - RX INT FLAGS (BIT 15 SET IF RX.DATA.AVAIL IS CLEAR).
4463      ; *
4464      ; * CALLING SEQUENCE:  PUT THE ADDRESS OF THE LABEL RXINPT IN THE VECTOR
4465      ; * LOCATION.
4466      ; *
4467      ; * COMMENTS:     IN CASE OF OVERFLOW OF THE MEMORY BUFFER, BU*PTR WILL BE
4468      ; *                MAINTAINED EQUAL TO BUFEND AND THE WORD AT BU*PTR WILL BE
4469      ; *                THE LAST WORD READ FROM THE DUT FIFO.
4470      ; *                NOTE: THIS ROUTINE CAN DESTROY TX ACTIONS BY READING THE CSR.
4471      ; *
4472      ; * SUBORDINATE ROUTINES CALLED: NONE.
4473      ; * - - - - -
4474
4475      RXINPT:: SAVE                                ;SAVE CONTENTS OF GPRS R0 THRU R5.
4476      ;R5,PREG05                                ;CALL REGISTER SAVE SUBRT.
4477      MOV     @CSRA,R1                            ;READ THE CONTENTS OF THE CSR.
4478      BIT     @BIT7,R1                            ;TEST RX.DATA.AVAIL BIT.
4479      BNE     2$                                  ;BRANCH AROUND SETTING FLAG IF BIT IS SET.
4480      BIS     @BIT15,RXINTF                       ;SET THE RX.DATA.AVAIL CLEAR FLAG.
4481      MOV     RXINTC,R1                           ;GET THE INTERRUPT COUNT.
4482      INC     R1                                  ;INCREMENT THE COUNT.
4483      BEQ     4$                                  ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
4484      MOV     R1,RXINTC                           ;SAVE NEW COUNT VALUE.
4485      MOV     BU*PTR,R2                           ;GET THE POINTER TO NEXT FREE BUFFER WORD.
4486      MOV     @RBUFA,(R2)+                        ;READ A CHAR FROM THE FIFO INTO BUFFER.
4487      CMP     R2,BUFEND                           ;TEST FOR POINTER BEYOND END OF BUFFER.
4488      BHS     60$                                  ;SKIP THE PTR UPDATE IF PTR OUT OF BOUNDS.
4489      MOV     R2,BU*PTR                           ;UPDATE THE BUFFER POINTER.
4490      PASS                                         ;RESTORE GPRS.
4491      JSR     PC,@(SP)+                             ;RETURN TO PREG05 SUBRT.
4492      RTI

```

```

4492 .SBTTL GLOBAL TRAP SERVICE ROUTINE - TP4RTN -
4493 ;*****
4494 ;* BUS TIME-OUT TRAP (004 TRAP) SERVICE ROUTINE -
4495 ;* THIS ROUTINE DETERMINES IF THE 004 TRAP WAS CAUSED BY
4496 ;* AN "EXPECTED" ERROR OR NOT BY EXAMINING THE RETURN PC VALUE ON THE
4497 ;* STACK. IF THE TRAP IS UNEXPECTED, THIS ROUTINE JUMPS TO THE NORMAL
4498 ;* DIAGNOSTIC SUPERVISOR 004 TRAP HANDLING ROUTINE.
4499 ;*
4500 ;*
4501 ;* INPUTS: SP - POINTS TO THE PC WHERE THE TRAP OCCURED.
4502 ;* ADRPTR - LABEL AT THE ADDRESS WHERE "EXPECTED" TRAPS OCCUR.
4503 ;* TP4FLG - 004 TRAP FLAGS.
4504 ;*
4505 ;* OUTPUTS: TP4FLG - BIT 15 IS SET IF "EXPECTED" TRAP OCCURED.
4506 ;*
4507 ;* CALLING SEQUENCE: PUT ADDRESS POINTED TO BY TP4RTN IN 004 VECTOR.
4508 ;* OCCURENCE OF 004 TRAP VECTORS TO THIS ROUTINE.
4509 ;*
4510 ;* COMMENTS: ANY 004 TRAP WHICH OCCURS AT AN ADDRESS OTHER THAN THAT LABELED
4511 ;* ADRPTR WILL BE HANDLED BY THE NORMAL 004 TRAP SERVICE ROUTINE.
4512 ;*
4513 ;* SUBORDINATE ROUTINES CALLED: NONE.
4514 ;*****
4515
4516 024124 021627 017764 TP4RTN;; CMP (SP),ADRPTR ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
4517 024130 001402 BEQ 2$ ;IF THEY MATCH, CONTINUE THIS ROUTINE.
4518 024132 000177 156164 JMP @TP4VEC ;IF NOT,JUMP TO NORMAL 004 TRAP SERVICE RIN.
4519 024136 052767 100000 156154 2$; BVS @BIT15,TP4FLG ;SET THE 004 TRAP OCCURED FLAG.
4520 024144 000002 RTI ;ALL DONE, GO BACK TO THE TEST.

```

```

4522          SBTTL  INTERUPT SERVICE ROUTINE      TXINTR
4523          ;* *****
4524          ;* TRANSMIT INTERRUPT SERVICE ROUTINE
4525          ;* THIS ROUTINE HANDLES A TRANSMIT INTERRUPT FROM THE DEVICE UNDER TEST
4526          ;* (DUT) BY COUNTING THE INTERRUPT AND READING THE DUT CSR TO CLEAR THE
4527          ;* INTERRUPT REQUEST. THIS ROUTINE ALSO SETS A FLAG TO INDICATE THAT
4528          ;* A TX INTERRUPT HAS OCCURRED AND SETS A FLAG IF THE TX.ACTION BIT IS
4529          ;* NOT SET IN THE READ CONTENTS OF THE DUT CSR.
4530          ;*
4531          ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR.
4532          ;*              TXINTC - HOLDS THE COUNT OF THE NUMBER OF TX INTERUPTS.
4533          ;*              TXINTF - TX INTERRUPT FLAGS.
4534          ;*
4535          ;* OUTPUTS:     TXINTC - CONTAINS THE UPDATED TX INTERRUPT COUNT.
4536          ;*              TXINTF - TX INT FLAGS (BIT 0 SET, BIT 15 SET IF TX.ACTION CLR).
4537          ;*
4538          ;* CALLING SEQUENCE: PUT THE ADDRESS OF THE LABEL TXINTR IN THE VECTOR
4539          ;* LOCATION.
4540          ;*
4541          ;* COMMENTS:
4542          ;*
4543          ;* SUBORDINATE ROUTINES CALLED: NONE
4544          ;* *****
4545
4546          TXINTR:: SAVE                                ;SAVE CONTENTS OF GPRS R0 THRU R5.
4547          ;*          JSR                                R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4548          ;*          MOV TXINTC,R1                    ;GET THE TX INTERRUPT COUNT.
4549          ;*          INC R1                            ;INCREMENT THE COUNT.
4550          ;*          BVC R1                            ;BRANCH IF NO OVERFLOW OCCURRED.
4551          ;*          DEC R1                            ;RESET THE COUNT TO 177777.
4552          ;*          2$: MOV R1,TXINTC                ;SAVE NEW COUNT VALUE.
4553          ;*          MOV TXINTF,R3                    ;GET THE TX INTERRUPT FLAGS.
4554          ;*          MOV @CSRA,R2                     ;READ THE CSR.
4555          ;*          BMI 4$                             ;SKIP SETTING OF FLAG IF TX.ACTION IS SET.
4556          ;*          BIS @BIT15,R3                   ;SET THE TX.ACTION CLEAR FLAG.
4557          ;*          4$: BIS @BIT0,R3                 ;SET THE TX INT HAS OCCURRED FLAG.
4558          ;*          MOV R3,TXINTF                    ;UPDATE THE TX INTERRUPT FLAGS.
4559          ;*          60$: PASS                         ;RESTORE GPRS.
4559          ;*          ;*          JSR                                PC,@(SP). ;RETURN TO PREG05 SUBRT.
4559          ;*          RTI

```


C10

C:\DHU10.DHU.11\UNC.151.PART1 MACRO M1200 15-MAR-84 09:15 PAGE 93
INTERUPT SERVICE ROUTINE

SEQ 119

4568

4569

4570

4571

4572

4573

4574

4575

4576

4577 024222

024223

4578

4579 024223

024223 000167

024224 000000

4580

4581

4582

4583 024226

024226

024226 104425

.SBTTL REPORT CODING SECTION

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

BGNRPT

C:RPT::

EXIT RPT

.WORD C:JMP
.WORD C:10017-2

.EVEN

ENDRPT

L10017:
TRAP C:RPT

D10

620HUB0 D10 11 LUNC 1ST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 94

SEQ 120

```
PROTECTION TABLE

4585          .SBITL  PROTECTION TABLE
4593
4594
4595          ;
4596          ; THIS TABLE IS USED BY THE RUNTIME SERVICES
4597          ; TO PROTECT THE LOAD MEDIA.
4598          ;
4599
4600 024230          BGNPROT
4601          L $PROT::
4602 024230 177777          -1          ;OFFSET INTO P-TABLE FOR CSR ADDRESS
4603 024230 177777          -1          ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
4604 024234 177777          -1          ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
4605
4606 024236          ENDPROT
4607
```



```

4629
4630
4631 .SBTTL INITIALIZE SECTION
4632 ;**
4633 ;*****
4634 ;* THIS SECTION CONTAINS THE CODE WHICH IS PERFORMED AT THE BEGINNING OF
4635 ;* EACH PASS OR AFTER A CONTINUE COMMAND.
4636 ;* THIS CODE PERFORMS THE FOLLOWING ACTIONS:
4637 ;*
4638 ;* MOVES THE INFORMATION HELD IN THE HARDWARE P-TABLE INTO THE GLOBAL
4639 ;* DATA AREA.
4640 ;*
4641 ;*****
4642 ;--
4643 024236 BGNINIT
4644 024236 L$INIT::
4645 ;SEE IF PROGRAM JUST STARTED, BR IF YES
4646 024236 012700 000040 MOV $EF.START,RO
4647 024242 104447 TRAP C$REFG
4648 024244 BCOMPLETE NEWSTA
4649 024244 103416 BCS NEWSTA
4650 ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
4651 024246 012700 000037 MOV $EF.RESTART,RO
4652 024252 104447 TRAP C$REFG
4653 024254 BCOMPLETE NEWRES
4654 024254 103556 BCS NEWRES
4655 ;SEE IF THIS IS A NEW PASS, BR IF YES
4656 024256 012700 000035 MOV $EF.NEW,RO
4657 024262 104447 TRAP C$REFG
4658 024264 BCOMPLETE NEWPAS
4659 024264 103555 BCS NEWPAS
4660 ;SEE IF PROGRAM WAS JUST CONTINUED
4661 024266 012700 000036 MOV $EF.CONTINUE,RO
4662 024272 104447 TRAP C$REFG
4663 024274 BNCOMPLETE GETPRM
4664 024274 103161 BCC GETPRM
4665 024276 000167 000540 JMP ENDIT
4666 024302 NEWSTA: BRESET ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
4667 024302 104433 TRAP C$RESET
4668 ;*
4669 ;* SET UP FOR LINE TIME CLOCK INTERRUPTS.
4670 ;*
4671 ;--
4672 024304 CLOCK L,R1 ;GET THE CLOCK PARAMETERS.
4673 024304 012700 000114 MOV $L,RO
4674 024310 104462 TRAP C$CLK
4675 024312 010001 MOV RO,R1
4676 024314 012167 156014 MOV (R1)+,CLKCSR ;STORE CLOCK CSR ADDRESS.
4677 024320 012167 156012 MOV (R1)+,CLKBRL ;STORE CLOCK BUS REQ INT LEVEL.
4678 024324 012167 156010 MOV (R1)+,CLKVEC ;STORE CLOCK INTERRUPT VECTOR.
4679 024330 012167 156006 MOV (R1)+,CLKHRZ ;STORE CLOCK FREQUENCY.
4680 024334 026727 156002 000062 CMP CLKHRZ,$50. ;TEST FOR 50HZ LINE FREQUENCY.
4681 024342 001004 BNE 2$ ;BRANCH IF CLOCK IS NOT 50HZ.

```

```

INITIALIZE SECTION

4669 024344 012767 000024 156002      MOV    #20.,MSTICK      ;INDICATE 20MS PER CLOCK TICK.
4670 024352 000403                    BR      4$
4671 024354 012767 000021 155772 2$:  MOV    #17.,MSTICK      ;INDICATE 17 MS PER CLOCK TICK.
4672 024362                    4$:  SETVEC  CLKVEC,#CLKINT,#PRI06 ;INITIALIZE CLOCK INTERRUPT VECTOR.
                                MOV    #PRI06,-(SP)
                                MOV    #CLKINT,-(SP)
                                MOV    CLKVEC,-(SP)
                                MOV    #3,-(SP)
                                TRAP    C$SVEC
                                ADD     #10,SP
4673 024410 016700 155726      MOV    CLKHRZ,R0      ;INITIALIZE THE BREAK COUNT
4674 024414 006300              ASL     R0              ; TO CAUSE A BREAK
4675 024416 010067 155730      MOV    R0,BCOUNT      ; EVERY 2 SECONDS.
4676 024422                    SETPRI  #PRI05          ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
                                MOV     #PRI05,R0
                                TRAP    C$SPRI
                                024422 012700 000240
                                024426 104441

4677
4678
4679
4680
4681
4682 024430 016767 153350 155664      MOV    4,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
4683 024436 012767 024124 153340      MOV    #TP4RTN,4    ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4684
4685
4686
4687 024444 005067 155650          CLR    TP4FLG      ;CLEAR THE 004 TRAP FLAG.
4688 024450 012767 000100 155654      MOV    #BIT6,WORD1    ;SET UP TO SET BIT6 OF THE LTC CSR.
4689 024456 012700 002332          MOV    #WORD1,R0      ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
4690 024462 016701 155646          MOV    CLKCSR,R1      ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
4691 024466 004767 173260          JSR    PC,CKTRAP      ;MOVE AND CHECK FOR TRAP.
4692 024472 016767 155624 153304      MOV    TP4VEC,4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
4693 024500 103403              BCS     6$              ;IF NO TRAP, LTC IS THERE SO CONTINUE.
4694 024502 005067 155634          CLR    CLKHRZ      ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
4695 024506 000402              BR      8$              ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
4696
4697
4698
4699 024510 004767 173012          JSR    PC,CALMSL
4700
4701
4702
4703
4704 024514 016767 153264 155600      MOV    4,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
4705 024522 012767 024124 153254      MOV    #TP4RTN,4    ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4706 024530 005067 155564          CLR    TP4FLG      ;CLEAR THE 004 TRAP FLAG.
4707 024534 005067 155572          CLR    WORD1      ;PREPARE TO CLEAR THE MEM MGT SRO REGISTER.
4708 024540 012700 002332          MOV    #WORD1,R0      ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
4709 024544 016701 155610          MOV    MMSRO,R1      ;SELECT MEM MGT SRO REGISTER AS DESTINATION.
4710 024550 005067 155606          CLR    MMPRES      ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
4711 024554 005067 155604          CLR    MMENAB      ;INDICATE MEM MGT IS NOT ENABLED.
4712 024560 004767 173166          JSR    PC,CKTRAP      ;CLEAR THE MEM MGT SRO REG AND CHECK FOR TRAP.
4713 024564 016767 155532 153212      MOV    TP4VEC,4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
4714 024572 103003              BCC     10$             ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
4715 024574 012767 000001 155560      MOV    #1,MMPRES      ;INDICATE THAT MEM MGT IS PRESENT.
4716 024602 005067 155504          CLR    PASCNT      ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4717 024606 000167 000006          JMP     NEWPAS      ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.

```

```

4718
4719 024612          NEWRES: BRESET          ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
      024612 104433          TRAP C$RESET
4720 024614 005067 155472          CLR      PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4721 024620
4722 024620 012767 177777 155410  NEWPAS: MOV      #-1,UNITN      ;RESET LOGICAL DEVICE TO -1
4723
4724          ;* INCREMENT THE PASS COUNTER, CORRECT FOR ANY OVERFLOW.
4725          ; THIS COUNTER IS USED IN THE ROM VERSION TEST.
4726          ;*
4727 024626 005267 155460          INC      PASCNT          ;INCREMENT THE PASS COUNTER.
4728 024632 001002          BNE      GETPRM      ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
4729 024634 005367 155452          DEC      PASCNT          ;SET PASS COUNT TO 177777 OCTAL.
4730
4731          ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
4732 024640          GETPRM:
4733 024640 005267 155372          INC      UNITN          ;INCREMENT LOGICAL DEVICE NUMBER
4734 024644 026767 155366 155140  CMP      UNITN,L$UNIT      ;SEE IF MAXIMUM UNIT NO. EXCEEDED
4735 024652 002362          BGE      NEWPAS          ;BR IF YES
4736
4737 024654          GPWARD UNITN,R1          ;GET P-TABLE POINTER INTO R1
      024654 016700 155356          MOV      UNITN,R0
      024660 104442          TRAP C$GPWARD
      024662 010001          MOV      R0,R1
4738 024664          BCOMPLETE 30$          ;BR IF DEVICE AVAILABLE
      024664 103401          BCS      30$
4739 024666 000764          BR      GETPRM          ;SKIP THIS DEVICE
4740
4741
4742          ;***** HARDWARE PARAMETER MOVING CODE *****
4743 024670 012167 155346 30$: MOV      (R1)+,CSRA      ;STORE DHU-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
4744 024674 012102          MOV      (R1)+,R2          ;GET THE RX INTERRUPT VECTOR ADDRESS.
4745 024676 010267 155330          MOV      R2,RXVECA      ;STORE RX INT VECTOR ADDRESS.
4746 024702 062702 000004          ADD      #4,R2          ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
4747 024706 010267 155322          MOV      R2,TXVECA      ;STORE TX INT VECTOR ADDRESS.
4748 024712 012167 155312          MOV      (R1)+,ACTLNS      ;STORE DHU-11 ACTIVE LINE BIT MAP
4749 024716 111167 155316          MOV      (R1),BRLEVL      ;STORE DHU-11 INTERRUPT BUS REQUEST LEVEL
4750
4751          ;*
4752          ; CALCULATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
4753          ; DEVICE REGISTER ADDRESS TABLE.
4754          ;*
4754 024722 016701 155314          MOV      CSRA,R1          ;COPY CSR ADDRESS
4755 024726 005201          INC      R1          ;INCREMENT CSR ADDRESS
4756 024730 005201          INC      R1          ; COPY BY 2.
4757 024732 012703 000007          MOV      #7,R3          ;SET UP REGISTER COUNT
4758 024736 012702 002244          MOV      #RBUFA,R2      ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
4759 024742 010122          12$: MOV      R1,(R2)+      ;STORE REGISTER ADDRESS IN TABLE
4760 024744 005201          INC      R1          ;INCREMENT REGISTER ADDRESS
4761 024746 005201          INC      R1          ; BY 2, FOR THE NEXT DEVICE REGISTER.
4762 024750 005303          DEC      R3          ;DECREMENT REGISTER COUNT
4763 024752 001373          BNE      12$          ;LOOP IF NOT DONE
4764
4765          ;*
4766          ; INITIALISE THE BMP CODE QUEUE.
4767          ;*
4768 024754 012700 002526          MOV      #BMPCQB,R0      ;GET THE START ADDRESS OF THE QUEUE.
4769 024760 012701 002726          MOV      #BMPCQE,R1      ;GET THE END ADDRESS OF THE QUEUE.

```

H10

CZDHUB0 DHU-11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 95-3
INITIALIZE SECTION

SEQ 124

```

4770 024764 010067 155534      MOV    R0,BMPCQP      ;SET THE POINTER TO THE START OF THE QUEUE.
4771 024770 005020      14$:  CLR    (R0)+      ;CLEAR OUT THE CONTENTS OF THE QUEUE.
4772 024772 020001      CMP    R0,R1      ;CHECK IF END OF QUEUE HAS BEEN REACHED.
4773 024774 103775      BLD    14$      ;LOOP IF NOT ALL DONE.
4774
4775      ;*
4776      ; REPORT THE UNIT NUMBER IF THE SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
4777      ; AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
4778 024776 032767 000020 155220      BIT    #BIT4,OPTION      ;CHECK IF THE QUESTION WAS ANSWERED YES.
4779 025004 001416      BEQ    16$      ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
4780 025006 026727 155000 000001      CMP    L$UNIT,#1      ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
4781 025014 003412      BLE    16$      ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
4782 025016      PRINTF  #MFUNIT,UNITN      ;REPORT UNIT NUMBER.
      MOV    UNITN,-(SP)
      MOV    #MFUNIT,-(SP)
      MOV    #2,-(SP)
      MOV    SP,R0
      TRAP   C$PNTF
      ADD    #6,SP
4783 025042      16$:
4784
4785 025042 005067 155240      ENDIT: CLR    CTRLCF      ;CLR THE CTRL-C TEST ABORT FLAG.
4786      ;*
4787      ; SET THE PROCESSOR PRIORITY TO DISABLE ALL INTERRUPTS.
4788      ;*
4789 025046      SETPRI  #PRI07      ;SET PROCESSOR PRIORITY TO 7.
      MOV    #PRI07,R0
      TRAP   C$SPRI
      MOV    025046 012700 000340
      MOV    025052 104441
4790 025054      ENDINIT
      L10021:
      TRAP   C$INIT
      MOV    025054 104411
4791
4792      INUM == 0      ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.

```

```

4801
4802
4803      .SBTTL  AUTODROP SECTION
4804
4805
4806      ;**
4807      ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
4808      ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
4809      ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
4810      ; DROPPED FROM TESTING.
4811      ;**
4812
4813 025056      BGNAUTO
4814      025056
4815
4816
4817
4818
4819
4820
4821
4822 025056      ENDAUTO
4823      025056
4824      025056 104461
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

```

L \$AUTO::

L10022: TRAP C\$AUTO

```

4831
4832
4833      .SBTTL  CLEANUP CODING SECTION
4834
4835      ;++
4836      ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4837      ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
4838      ;--
4839
4840 025060      BGNCLN
4841      025060
4842
4843      L$CLEAN::
4844
4845      1ST      CTRLCF      ;DID WE GET HERE BY CTRL-C FROM TEST?
4846      B1.Q     2$          ;CTRL-C FROM TEST? NO, SKIP BUS RESET.
4847      BRESET   2$          ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
4848
4849      2$:      TRAP      C$RESET
4850
4851      EXIT     CLN
4852
4853      TRAP     C$EXIT
4854      .WORD    L10023-.
4855
4856      .EVEN
4857
4858      ENDCLN
4859
4860      L10023:
4861      TRAP     C$CLEAN
4862
4863      025060 005767 155222
4864      025064 001401
4865      025066 104433
4866      025070
4867      025070 104432
4868      025072 000002
4869
4870      025074
4871      025074 104412

```


K10

CZDHUBC DHU 11 FUNC 1ST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 98
CLEANUP CODING SECTION

SEQ 127

```

4879
4880
4881      .SBTTL  DROP UNIT SECTION
4882
4883      ;++
4884      ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4885      ; TO NO LONGER BE TESTED.
4886      ;--
4887
4888      BGNDU
4897      025076      PRINTF #DROP,RO      ;REPORT UNIT THAT HAS BEEN DROPPED.
      025076      L$DU::
      025100      010046      025122      MOV      RO,-(SP)
      025104      012746      000002      MOV      #DROP,-(SP)
      025110      010600      MOV      #2,-(SP)
      025112      104417      MOV      SP,RO
      025114      062706      000006      TRAP     C$PNTF
      025120      000427      ADD      #6,SP
4898      BR      EDROP      ;BRANCH AROUND THE MESSAGE.
4899
4900      025122      045      101      040      DROP:  .ASCIZ/%A UNIT#D6%A DROPPED FROM FURTHER TESTING.%N/
      025125      125      116      111
      025130      124      045      104
      025133      066      045      101
      025136      040      104      122
      025141      117      120      120
      025144      105      104      040
      025147      106      122      117
      025152      115      040      106
      025155      125      122      124
      025160      110      105      122
      025163      040      124      105
      025166      123      124      111
      025171      116      107      056
      025174      045      116      000
4901
4902      025200      EDROP:  .EVEN
4903      025200      EXIT      DU
      025200      000167
      025202      000000
      .WORD      J$JMP
      .WORD      L10024-2-.
4904
4905
4906
4907      025204      ENDDU
      025204
      025204      104453
      L10024:
      TRAP      C$DU

```

```

4916
4917
4918 .SBTTL ADD UNIT SECTION
4919
4920 ;**
4921 ; THE ADD UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
4922 ; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
4923 ; TO THE TEST CYCLE.
4924 ;--
4925
4926 025206 BGNAU
4927 025206 L$AU::
4928 ;*****
4929 ; INSERT ADD CODE HERE. THIS CODE WILL BE EXECUTED AFTER
4930 ; AN "ADD" COMMAND. THE PURPOSE OF THIS CODE IS TO DO ANY
4931 ; HOUSEKEEPING THAT MAY BE NECESSARY AFTER A UNIT HAS BEEN ADDED.
4932 ; THIS SECTION IS OPTIONAL.
4933 ;*****
4934
4935 025206 EXIT AU
4936 025206 000167 .WORD J$JMP
4937 025210 000000 .WORD L10025-2-.
4938
4939 .EVEN
4940 025212 ENDAU
4941 025212 104452 L10025: TRAP C$AU

```



```

4943 ,SBTTL  HARDWARE TEST          - ADRA -
4944 ;**
4945 ;*****
4946 ;*                                - REGISTER ADDRESS TEST -
4947 ;*
4948 ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS WILL RESPOND TO THE PROPER
4949 ;* UNIBUS HANDSHAKING SIGNALS WHEN ACCESSED. IF THE DHU11 DOES NOT RESPOND
4950 ;* TO THE ACCESS ATTEMPTS (IF THE DHU11 IS AT THE WRONG ADDRESS, FOR EXAMPLE)
4951 ;* THE 004 BUS TIME-OUT TRAP IS DETECTED BY THIS ROUTINE AND AN ERROR
4952 ;* IS REPORTED. THIS TEST IS PERFORMED ON LINE 0 ONLY.
4953 ;*
4954 ;*****
4955 ;--
4956
4957 025214      BGNTST
4958 025214      TNUM = TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
4959 025214 012767 000001 155102      MOV  @TNUM,ISTNUM      ;SET UP THE TEST NUMBER. (1)
4960 025222 012767 177777 155056      MOV  @-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
4961 025230 012767 000145 156532      MOV  @101,ERRNBR      ;SET THE TEST ERROR NUMBER IN THE TABLE.
4962 025236 012767 005560 156526      MOV  @EM0103,ERRMSG      ;SET UP THE TEST FAILURE MESSAGE IN THE TABLE.
4963 025244 012767 015510 156522      MOV  @ERO101,ERRBLK      ;SET UP THE ERROR ROUTINE IN THE ERROR TABLE.
4964
4965 ; SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
4966 ;--
4967 025252 016767 152526 155042      MOV  4,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
4968 025260 012767 024124 152516      MOV  @TP4RTN,4      ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4969 025266 005005                      CLR  R5      ;CLEAR THE ERROR FLAGS.
4970
4971 ;
4972 ; HERE BEGINS THE LOOP TO TEST THE REGISTERS FOR A LINE.
4973 ; FIRST TEST THE CSR AND SET THE IND.ADR.REG (I.A.R) FIELD.
4974 ;--
4975 025270 016700 154746      MOV  CSRA,R0      ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
4976 025274 012701 025466      MOV  @52$,R1      ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
4977 025300 004767 172446      JSR  PC,CKTRAP      ;MOVE AND CHECK FOR TRAP.
4978 025304 103402              BCS  4$      ;IF NO TRAP, BYPASS ERROR.
4979 025306 052705 100001      BIS  @100001,R5      ;SET FATAL READ ERROR FLAGS.
4980 025312 042767 000017 000146 4$: BIC  @17,52$      ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
4981 025320 010100              MOV  R1,R0      ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
4982 025322 016701 154714      MOV  CSRA,R1      ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
4983 025326 004767 172420      JSR  PC,CKTRAP      ;MOVE AND CHECK FOR TRAP.
4984 025332 103403              BCS  6$      ;IF NO TRAP, BYPASS ERROR.
4985 025334 052705 100002      BIS  @100002,R5      ;SET FATAL WRITE ERROR FLAGS.
4986 025340 000434              BR   40$      ;EXIT AND REPORT FATAL ERROR.
4987
4988 ;
4989 ; NOW, WE TEST EACH REGISTER FOR THIS LINE.
4990 ;--
4990 025342 012702 000010      6$: MOV  @8$,R2      ;INIT REGISTER COUNTER TO 8.
4991 025346 016767 154670 000110      MOV  CSRA,50$      ;INITIALIZE THE REGISTER POINTER.
4992 025354 016700 000104      8$: MOV  50$,R0      ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
4993 025360 012701 025466      MOV  @52$,R1      ;SET UP LOCAL STORAGE AS THE DEST FOR CKTRAP.
4994 025364 004767 172362      JSR  PC,CKTRAP      ;PERFORM THE MOVE, CHECK FOR TRAP.
4995 025370 103402              BCS  10$      ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
4996 025372 052705 100001      BIS  @100001,R5      ;SET FATAL READ ERROR FLAGS.
4997 025376 010100              10$: MOV  R1,R0      ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
4998 025400 016701 000060      MOV  50$,R1      ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.

```

```

4999 025404 004767 172342      JSR      PC,CKTRAP      ;PERFORM THE MOVE, CHECK FOR TRAP.
5000 025410 103402      BCS      12$      ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
5001 025412 052705 100002      BIS      0100002,R5      ;SET FATAL WRITE ERROR FLAGS.
5002 025416 005267 000042      12$: INC      50$      ;INCREMENT THE REGISTER
5003 025422 005267 000036      INC      50$      ; POINTER BY 2.
5004 025426 005302      DEC      R2      ;COUNT THE REGISTER.
5005 025430 001351      BNE      8$      ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
5006
5007
5008      ;+
5009      ; DONE CHECKING DEVICE REGISTER ADDRESSES.
5010      ; REPORT ANY ERRORS AND EXIT.
5011      ;-
5012 025432 016767 154664 152344 40$: MOV      TP4VEC,4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
5013 025440 005705      TST      R5      ;CHECK THE ERROR FLAGS.
5014 025442 100012      BPL      60$      ;EXIT ROUTINE IF NO ERRORS.
5015
5016      ;+
5017      ; REPORT "DEVICE REGISTER ACCESS TEST FAILED"
5018      ;-
5018 025444      ERROR
5019 025444 104460      TRAP      C$ERROR
5020
5021 025446      DODU      UNITN      ;DROP THIS UNIT FROM FUTHER TESTING.
5022 025446 016700 154564      MOV      UNITN,R0
5023 025452 104451      TRAP      C$DODU
5024 025454 005067 154626      CLR      CTRLCF      ;INDICATE NO CTRL-C ABORT FROM TEST.
5025 025460      DOCLN      ;ABORT THIS SUB PASS.
5026 025460 104444      TRAP      C$DCLN
5027 025462 000402      BR      60$      ;
5028
5029      ;***** LOCAL STORAGE. *****
5030      50$: .WORD 0      ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
5031      52$: .WORD 0      ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
5032      ;***** END *****
5033 025470 005067 154612      60$: CLR      CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5034 025474      ENDTST
5035 025474 104401      L10026: TRAP      C$ETST

```

```

5034 .SBTTL  HARDWARE TEST - MRSTA
5035 *****
5036 - MASTER RESET WITH SELFTEST TEST
5037 THIS TEST VERIFIES THAT THE MASTER RESET BIT WILL CLEAR AFTER A DEVICE
5038 RESET AND THE PERFORMANCE OF THE OUT ROM BASED SELFTEST.
5039 *****
5040
5041 025476 BGNTEST
5042 025476 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5043 025476 012767 000002 154620 MOV @TNUM,TSTNUM ;SET UP THE TEST NUMBER. (2)
5044 025504 012767 177777 154574 MOV @1,CIRLCF ;INDICATE THAT WE ARE IN A TEST.
5045 025512 SETPRI @PRI05 ;ALLOW LTC INTERRUPTS.
5046 025512 012700 000240 MOV @PRI05,R0
5047 025516 104441 TRAP C$SPRI
5048 025520 012767 000001 156240 MOV @1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5049 025526 012767 005616 156236 MOV @EM0201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5050 025534 012767 016042 156232 MOV @ER0201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5051
5052 ; WAIT UP TO 5 SECONDS FOR THE OUT MASTER RESET BIT TO CLEAR.
5053
5054 MOV @5000.,R1 ;TIME OUT VALUE IS 5.0 SECONDS.
5055 MOV @BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5056 CLR R3 ;WAITING FOR BIT TO CLEAR.
5057 MOV CSRA,R4 ;BIT IS IN THE OUT'S CSR.
5058 JSR PC,MSLGET ;WAIT FOR OUT CSR MR BIT TO CLEAR.
5059 BCS 21 ;SKIP TO RESET OUT IF MR CLEAR.
5060
5061 ; OUT MASTER RESET BIT DID NOT GO CLEAR. DEVICE MAY BE STUCK IN SOME
5062 ; ODD STATE. TRY TO RESET DEVICE WITH A BUS RESET.
5063
5064 BRESET ;NO, TRY TO JOG DEVICE WITH BUS RESET.
5065 JSR PC,SKPSTS ;TRY TO SKIP THE SELFTEST.
5066 MOV @5000.,R1 ;TIME OUT VALUE IS 5.0 SECONDS.
5067 JSR PC,MSLGET ;WAIT FOR OUT CSR MR BIT TO CLEAR.
5068 BCC 41 ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
5069
5070 ; SET THE MASTER RESET BIT AND VERIFY THAT IT CLEARS WITHIN THE PROPER TIME.
5071
5072 MOV @5000.,R1 ;TIME OUT VALUE IS 5.0 SECONDS.
5073 MOV R2,(R4) ;SET THE OUT MASTER RESET BIT.
5074 JSR PC,MSLGET ;WAIT FOR OUT CSR MR BIT TO CLEAR.
5075 BCC 41 ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
5076 MOV @5000.,R1
5077 SUB R1,R1 ;CALCULATE # OF MS FOR MR TO CLEAR.
5078 BEQ 41 ;GO REPORT ERROR IF MR CLEAR IMMEDIATELY.
5079 CMP R1,@500.
5080 BLT 41 ;GO REPORT ERROR IF MR CLEAR IN < 1.0 SECOND.
5081 BR 601 ;EXIT THE TEST WITHOUT ERROR.
5082
5083 ; ERROR REPORT:
5084
5085 ;REPORT MR BIT WOULD NOT CLEAR AFTER A OUT RESET.
5086 MOV @01.,ERRMR ;SET THE ERROR NUMBER IN ERROR TABLE.
5087 MOV @EM0202,R1 ;SELECT ERROR MESSAGE.
5088 ERROR ;REPORT ERROR. ***** ERROR 0201 *****

```

C11

CZDHUBO DEDU 11 FUNC TST PART1 MACRO M1200 15-MAR 84 09:15 PAGE 101 1

SEQ 132

```

5087 025654 104460
5088 025656 000415
5089
5090 025660 012767 000312 156102 63: ;REPORT MR BIT CLEAR IMMEDIATELY AFTER DUT RESET.
5091 025666 012701 006024 ;MOV 0202,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5092 025672 104460 ;MOV 0EM0203,R1 ;SELECT ERROR MESSAGE.
5093 025674 000406 ;ERROR ;REPORT ERROR. >>>> ERROR 0202 <<<<
5094 ;TRAP C$ERROR
5095
5096 025676 012767 000313 156064 83: ;REPORT MR CLEAR WITHIN 1/2 SECOND OF DUT RESET.
5097 025704 012701 006167 ;MOV 0203,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5098 025710 104460 ;MOV 0EM0204,R1 ;SELECT ERROR MESSAGE.
5099 ;ERROR ;REPORT ERROR. >>>> ERROR 0203 <<<<
5100 ;TRAP C$ERROR
5101 025712 012700 000340 603: SETPRI 0PRI07 ;DISABLE ALL INTERRUPTS.
5102 025716 104441 ;MOV 0PRI07,R0
5103 025720 005067 154362 ;TRAP C$SPRI
5104 025724 104401 ;CLR CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
5105 025724 104401 ;ENDTST
5106 ;L10027:
5107 ;TRAP C$ETST

```

```

5104 .SBTTL  HARDWARE TEST - MRSSTA
5105 *****
5106 - MASTER RESET WITH SKIP SELFTEST TEST
5107 THIS TEST VERIFIES THAT THE MASTER RESET BIT WILL CLEAR AFTER A DEVICE
5108 RESET AND THE SKIPPING OF THE OUT ROM BASED SELFTEST.
5109 *****
5110
5111 025726 BGNTEST
5112 025726
5113 000003 TNUM = TNUM + 1 ; INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5114 025726 012767 000003 154370 MOV @TNUM,ISTNUM ; SET UP THE TEST NUMBER. (3)
5115 025734 012767 177777 154344 MOV @1,CTRLCF ; INDICATE THAT WE ARE IN A TEST.
5116 025742 012700 000240 SETPRI @PRI05 ; ALLOW LTC INTERRUPTS.
5117 025746 104441 MOV @PRI05,R0
5118 025750 012767 000001 156010 TRAP C$SPRI
5119 025756 012767 006346 156006 MOV @1,ERRTYP ; SET ERROR TYPE AS FATAL IN ERROR TABLE.
5120 025764 012767 016042 156002 MOV @EM0301,ERRMSG ; SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5121 MOV @ERO201,ERRBLK ; SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5122 ; WAIT UP TO 5 SECONDS FOR THE OUT MASTER RESET BIT TO CLEAR.
5123
5124 MOV @5000,R1 ; TIME-OUT VALUE IS 5.0 SECONDS.
5125 MOV @BIT05,R2 ; WAITING FOR MASTER RESET BIT.
5126 CLR R3 ; WAITING FOR BIT TO CLEAR.
5127 MOV CSRA,R4 ; BIT IS IN THE OUT S CSR.
5128 JSR PC,MSLGET ; WAIT FOR OUT_CSR_MR BIT TO CLEAR.
5129 BCS 2$ ; SKIP TO RESET OUT IF MR CLEAR.
5130 ; OUT MASTER RESET BIT DID NOT GO CLEAR. DEVICE MAY BE STUCK IN SOME
5131 ; ODD STATE. TRY TO RESET DEVICE WITH A BUS RESET.
5132
5133 BRESET ; NO. TRY TO JOG DEVICE WITH BUS RESET.
5134 026016 104433 TRAP C$RESET
5135 026020 004767 174320 JSR PC,SKPSTS ; TRY TO SKIP THE SELFTEST.
5136 026024 012701 011610 MOV @5000,R1 ; TIME-OUT VALUE IS 5.0 SECONDS.
5137 026030 004767 172130 JSR PC,MSLGET ; WAIT FOR OUT_CSR_MR BIT TO CLEAR.
5138 026034 103024 BCC 6$ ; GO REPORT ERROR IF MR BIT DID NOT CLEAR.
5139
5140 ; SET THE MASTER RESET BIT, TRY TO SKIP THE SELFTEST, AND VERIFY THAT THE
5141 ; MR BIT CLEARS WITHIN 1/5 SECOND.
5142
5143 2$: MOV @200,R1 ; TIME-OUT VALUE IS 1/5 SECOND.
5144 MOV R2,(R4) ; SET THE OUT MASTER RESET BIT.
5145 JSR PC,SKPSTS ; TRY TO SKIP THE SELFTEST.
5146 JSR PC,MSLGET ; WAIT FOR OUT_CSR_MR BIT TO CLEAR.
5147 BCC 4$ ; GO FIND OUT WHAT IS WRONG IF MR NOT CLEAR.
5148 MOV @200,R2
5149 SUB R1,R2 ; CALCULATE # OF MS FOR MR TO CLEAR.
5150 CMP R2,@10.
5151 BLT 8$ ; GO REPORT ERROR IF MR CLEAR IN < 10 MS.
5152 BR 60$ ; EXIT THE TEST WITHOUT ERROR.
5153
5154 ; MR DID NOT CLEAR WITHIN 1/5 SECOND. SEE IF IT CLEARS WITHIN 5 SECONDS.
5155
5156 4$: MOV @4800,R1 ; TIME-OUT VALUE IS 5 SECONDS MINUS 1/5 SECOND.
5157 JSR PC,MSLGET ; WAIT FOR OUT_CSR_MR BIT TO CLEAR.
5158 BCS 10$ ; GO REPORT ERROR IF MR CLEARED FINALLY.

```

E11

CZDHUB0 DHU-11 FUNC TEST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 102-1
HARDWARE TEST - MRSSTA

SEQ 134

```

5157
5158      ; ERROR REPORTS:
5159      ;
5160      ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5161 026106 012767 000455 155654 6$:      MOV    #0301.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5162 026114 012701 005651                MOV    #EM0202,R1      ;SELECT ERROR MESSAGE.
5163 026120                                ERROR                ;REPORT ERROR.          >>>>> ERROR #0301 <<<<<
                                104460                                TRAP    C$ERROR
5164 026122 000415                                BR      60$                ;EXIT THE TEST.
5165
5166      ;REPORT MR BIT CLEAR WITHIN 10 MS AFTER DUT RESET.
5167 026124 012767 000456 155636 8$:      MOV    #0302.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5168 026132 012701 006420                MOV    #EM0302,R1      ;SELECT ERROR MESSAGE.
5169 026136                                ERROR                ;REPORT ERROR.          >>>>> ERROR #0302 <<<<<
                                104460                                TRAP    C$ERROR
5170 026140 000406                                BR      60$                ;EXIT THE TEST.
5171
5172      ;REPORT MR CLEARED BETWEEN 1/5 SECOND AND 5 SECONDS OF DUT RESET.
5173 026142 012767 000457 155620 10$:     MOV    #0303.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5174 026150 012701 006560                MOV    #EM0303,R1      ;SELECT ERROR MESSAGE.
5175 026154                                ERROR                ;REPORT ERROR.          >>>>> ERROR #0303 <<<<<
                                104460                                TRAP    C$ERROR
5176
5177      60$:      SETPRI  #PRI07                ;DISABLE ALL INTERRUPTS.
                                000340                                MOV    #PRI07,RO
                                104441                                TRAP    C$SPRI
5178 026164 005067 154116                CLR    CTRLCF                ;INDICATE THAT WE COMPLETED THE TEST.
5179 026170                                ENDTST
                                104401                                L10030:
                                TRAP    C$ETST

```



```

5181 .SBTTL. HARDWARE TEST - RXCHRA -
5182 ;* *****
5183 ;* - RBUF REGISTER RX CHARACTER FIELD TEST -
5184 ;* THIS TEST VERIFIES THAT THE RX CHARACTER FIELD OF THE DUT RBUF REGISTER
5185 ;* APPEARS TO BE FUNCTIONING CORRECTLY. THIS TEST USES THE CODES WHICH
5186 ;* SHOULD BE IN THE FIFO AFTER A BOARD RESET AND SKIP SELFTEST SEQUENCE.
5187 ;* *****
5188 ;*
5189 026172 BGNST
5190 026172 T4::
026172 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
026172 012700 000240 MOV #PRI05,R0
026176 104441 TRAP C$SPRI
5191 000004 TNUM = TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5192 026200 012767 000004 154116 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (4)
5193 026206 012767 177777 154072 MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5194 026214 012767 000001 155544 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5195 026222 012767 006737 155542 MOV #EM0401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5196 026230 012767 016042 155536 MOV #ERO201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5197 ;*
5198 ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5199 ; AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5200 ;*
5201 026236 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5202 026242 012702 000040 MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5203 026246 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5204 026250 016704 153766 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5205 026254 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5206 026256 004767 174062 JSR PC,SKPSTS ;SKIP THE SELFTEST.
5207 026262 004767 171676 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5208 026266 103015 BCC 4$ ;GO REPORT ERROR IF MR DID NOT CLEAR.
5209 ;*
5210 ; READ 6 CHARACTERS FROM THE DUT AND VERIFY THAT THEY ARE VALID SELFTEST
5211 ; CODES.
5212 ;*
5213 026270 012400 MOV (R4)+,R0 ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
5214 026272 012701 000006 MOV #6,R1 ;INITIALIZE THE LOOP COUNTER.
5215 026276 011402 2$: MOV (R4),R2 ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
5216 026300 010200 MOV R2,R0
5217 026302 042700 177476 BIC #177476,R0 ;REMOVE ALL BUT BITS SPECIFIC TO SELFTEST CODE.
5218 026306 020027 000201 CMP R0,#201 ;CHECK THAT BITS 0,6, AND 7 ARE CORRECT.
5219 026312 001012 BNE 6$ ;GO REPORT ERROR IF CODE IS NOT SELFTEST CODE.
5220 026314 005301 DEC R1 ;COUNT THIS LOOP ITERATION.
5221 026316 001367 BNE 2$ ;LOOP IF NOT ALL LINES DONE.
5222 026320 000415 BR 60$ ;EXIT TEST, NO ERROR FOUND.
5223 ;*
5224 ;
5225 ; ERROR REPORTS:
5226 ;*
5227 ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5228 026322 012767 000621 155440 4$: MOV #0401.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5229 026330 012701 005651 MOV #EM0202,R1 ;SELECT ERROR MESSAGE.
5230 026334 ERROR ;REPORT ERROR. >>>>> ERROR #0401 <<<<<
026334 104460 TRAP C$ERROR
5231 026336 000406 BR 60$ ;EXIT THE TEST.
5232 ;
5233 ;REPORT IMPROPER CODE FOUND IN DUT RBUF AFTER RESET (SKIP SELFTEST).

```

G11

CZDHUB0 DHU-11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 103-1
HARDWARE TEST RXCHRA

SEQ 136

```

5234 026340 012767 000622 155422 6$:      MOV      #0402.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5235 026346 012701 007015                MOV      #EM0402,R1      ;SELECT ERROR MESSAGE.
5236 026352                ERROR                                ;REPORT ERROR.      >>>> ERROR #0402 <<<<<
                    026352 104460                                TRAP      C$ERROR
5237
5238 026354                60$:      SETPRI   #PRI07                ;DISABLE ALL INTERRUPTS.
                    026354 012700 000340                                MOV      #PRI07,R0
                    026360 104441                                TRAP      C$SPRI
5239 026362 005067 153720                CLR      CTRLCF                ;INDICATE THAT WE COMPLETED THE TEST.
5240 026366                ENDTST                                L10031:
                    026366 104401                                TRAP      C$ETST

```



```

5242 .SBTTL  HARDWARE TEST - RXFFDA -
5243 ;* *****
5244 ;* - RBUF REGISTER RX FLAG FIELD TEST -
5245 ;* THIS TEST VERIFIES THAT THE FIELD OF 3 FLAG BITS IN THE RBUF READS
5246 ;* AS ALL ONES WHEN THE SELFTEST CODES ARE BEING READ FROM THE DUT
5247 ;* AFTER A BOARD RESET AND SKIP SELFTEST SEQUENCE.
5248 ;*
5249 ;* *****
5250 BGNIST
5251 026370 012700 000240          T5:;
026370          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
026374 104441          MOV      #PRI05,R0
026374          TRAP    C$SPRI
5252          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5253 026376 012767 000005 153720  MOV      #TNUM,TSTNUM          ;SET UP THE TEST NUMBER. (5)
5254 026404 012767 177777 153674  MOV      #-1,CTRLCF          ;INDICATE THAT WE ARE WITHIN A TEST.
5255 026412 012767 000001 155346  MOV      #1,ERRIYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5256 026420 012767 007165 155344  MOV      #EM0501,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5257 026426 012767 016042 155340  MOV      #ER0201,ERRBLK          ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5258 ;*
5259 ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5260 ; AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5261 ;*
5262 026434 012701 011610          MOV      #5000.,R1          ;TIME-OUT VALUE IS 5.0 SECONDS.
5263 026440 012702 000040          MOV      #BIT05,R2          ;WAITING FOR MASTER RESET BIT.
5264 026444 005003          CLR      R3          ;WAITING FOR BIT TO CLEAR.
5265 026446 016704 153570          MOV      CSRA,R4          ;BIT IS IN THE DUT'S CSR.
5266 026452 010214          MOV      R2,(R1)          ;SET THE DUT MASTER RESET BIT.
5267 026454 004767 173664          JSR      PC,SKPSTS          ;SKIP THE SELFTEST.
5268 026460 004767 171500          JSR      PC,MSLGET          ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5269 026464 103013          BCC      4$          ;GO REPORT ERROR IF MR DID NOT CLEAR.
5270 ;*
5271 ; READ 8 CHARACTERS FROM THE DUT AND VERIFY THAT ALL 3 RX ERROR FLAGS ARE
5272 ; SET FOR EACH CHARACTERS.
5273 ;*
5274 026466 012400          MOV      (R4)+,R0          ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
5275 026470 012701 000010          MOV      #8.,R1          ;INITIALIZE THE LOOP COUNTER.
5276 026474 011402          2$: MOV      (R4),R2          ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
5277 026476 012700 070000          MOV      #70000,R0
5278 026502 040200          BIC      R2,R0          ;CALCULATE BIT MAP OF CLEAR RX ERROR FLAGS.
5279 026504 001012          BNE      6$          ;GO REPORT ERROR IF NOT ALL RX ERROR FLAGS SET.
5280 026506 005301          DEC      R1          ;COUNT THIS LOOP ITERATION.
5281 026510 001371          BNE      2$          ;LOOP IF NOT ALL LINES DONE.
5282 026512 000415          BR       60$          ;EXIT TEST, NO ERROR FOUND.
5283 ;*
5284 ; ERROR REPORTS:
5285 ;
5286 ;
5287 ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5288 026514 012767 000765 155246 4$: MOV      #0501.,ERRNBR          ;SET THE ERROR NUMBER IN ERROR TABLE.
5289 026522 012701 005651          MOV      #EM0202,R1          ;SELECT ERROR MESSAGE.
5290 026526          ERROR          ;REPORT ERROR.
5291 026530          BR       60$          ;EXIT THE TEST.
5292 ;
5293 ;REPORT ONE OR MORE RX ERROR FLAGS FOUND CLEAR WITH SELFTEST CODE.
5294 026532 012767 000766 155230 6$: MOV      #0502.,ERRNBR          ;SET THE ERROR NUMBER IN ERROR TABLE.

```



```

5302 .SBTTL  HARDWARE TEST - RDAA -
5303 ;* *****
5304 ;* - CSR RX DATA AVAILABLE BIT TEST -
5305 ;* THIS TEST VERIFIES THAT THE DUT CSR RX DATA AVAILABLE BIT IS SET BY THE
5306 ;* INCLUSION OF THE SELFTEST CODES IN THE DUT FIFO AND THAT THE BIT CLEARS
5307 ;* AFTER THE FIFO HAS BEEN EMPTIED.
5308 ;*
5309 ;* *****
5310 026562 BGNTEST
5311 026562 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T6::
5312 026562 012700 000240 MOV #PRI05,R0
5313 026562 104441 TRAP C$SPRI
5312 000006 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5313 026570 012767 000006 153526 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (6)
5314 026576 012767 177777 153502 MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5315 026604 012767 000001 155154 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5316 026612 012767 007575 155152 MOV #EM0601,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5317 026620 012767 016042 155146 MOV #ERO201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5318 ;*
5319 ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5320 ; AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5321 ;*
5322 026626 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5323 026632 012702 000040 MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5324 026636 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5325 026640 016704 153376 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5326 026644 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5327 026646 004767 173472 JSR PC,SKPSTS ;SKIP THE SELFTEST.
5328 026652 004767 171306 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5329 026656 103016 BCC 4$ ;GO REPORT ERROR IF MR DID NOT CLEAR.
5330 ;*
5331 ; CHECK THAT THE RX DATA AVAILABLE BIT IS SET.
5332 ;*
5333 026660 032714 000200 BIT #BIT7,(R4) ;TEST THE DUT RX.DATA.AVAIL BIT.
5334 026664 001422 BEQ 6$ ;GO REPORT ERROR IF BIT IS NOT SET.
5335 ;*
5336 ; READ CHARACTERS FROM THE DUT RX FIFO AND WAIT FOR RX.DATA.AVAIL TO GO CLEAR.
5337 ;*
5338 026666 012705 001130 MOV #600.,R5 ;ALLOW READING 600 CHARS BEFORE ERROR.
5339 026672 010403 MOV R4,R3
5340 026674 012300 MOV (R3)+,R0 ;CALCULATE THE RBUF ADDRESS.
5341 026676 011300 MOV (R3),R0 ;READ A CHARACTER FROM THE RX FIFO.
5342 026700 032714 000200 BIT #BIT7,(R4) ;TEST THE DUT RX.DATA.AVAIL BIT.
5343 026704 001427 BEQ 60$ ;EXIT TEST WITHOUT ERROR IF RX.DATA.AVAIL CLR.
5344 026706 005305 DEC R5 ;COUNT THE CHARACTER JUST READ.
5345 026710 001372 BNE 2$ ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
5346 026712 000416 BR 8$ ;GO REPORT ERROR IF RX.DATA.AVAIL WOULDN'T CLR.
5347 ;*
5348 ; ERROR REPORTS:
5349 ;*
5350 ;*
5351 ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5352 026714 012767 001131 155046 4$: MOV #0601.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5353 026722 012701 005651 MOV #EM0202,R1 ;SELECT ERROR MESSAGE.
5354 026726 104460 ERROR ;REPORT ERROR. >>>>> ERROR #0601 <<<<<
; TRAP C$ERROR

```

K11

CZDHUBO DHU-11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 105-1
HARDWARE TEST - RDAA -

SEQ 140

```

5355 026730 000415          BR      60$          ;EXIT THE TEST.
5356
5357          ;REPORT THAT RX.DATA.AVAIL BIT WAS NOT SET AFTER A RESET COMPLETION.
5358 026732 012767 001132 155030 6$:      MOV      #0602,,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5359 026740 012701 007637          MOV      #EM0602,R1      ;SELECT ERROR MESSAGE.
5360 026744          ERROR          ;REPORT ERROR.          >>>> ERROR #0602 <<<<<
          026744 104460          TRAP      C$ERROR
5361 026746 000406          BR      60$          ;EXIT THE TEST.
5362
5363          ;REPORT THAT RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO.
5364 026750 012767 001133 155012 8$:      MOV      #0603,,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5365 026756 012701 010017          MOV      #EM0603,R1      ;SELECT ERROR MESSAGE.
5366 026762          ERROR          ;REPORT ERROR.          >>>> ERROR #0603 <<<<<
          026762 104460          TRAP      C$ERROR
5367
5368 026764          60$:      SETPRI   #PRI07          ;DISABLE ALL INTERRUPTS.
          026764 012700 000340          MOV      #PRI07,R0
          026770 104441          TRAP      C$SPRI
5369 026772 005067 153310          CLR      CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5370 026776          ENDTST
          026776          L10033:
          026776 104401          TRAP      C$ETST

```

```

5372 .SETTL HARDWARE TEST - RDVA -
5373 ;* *****
5374 ;* - RBUF RX DATA VALID BIT TEST -
5375 ;* THIS TEST VERIFIES THAT THE DUT RBUF RX DATA VALID BIT IS SET BY THE
5376 ;* INCLUSION OF THE SELFTEST CODES IN THE DUT FIFO AND THAT THE BIT CLEARS
5377 ;* AFTER THE FIFO HAS BEEN EMPTIED.
5378 ;*
5379 ;* *****
5380 BGNTST
5381 027000 SETPRI 0PRI05 ;ALLOW LTC INTERRUPTS. T7:
      027000 MOV 0PRI05,R0
      027000 TRAP C$SPRI
      027004 104441
5382 000007 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5383 027006 012767 000007 153310 MOV 0TNUM,TSTNUM ;SET UP THE TEST NUMBER. (7)
5384 027014 012767 177777 153264 MOV 0-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5385 027022 012767 000001 154736 MOV 01,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5386 027030 012767 010202 154734 MOV 0EM0701,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5387 027036 012767 016042 154730 MOV 0ER0201,ERRSLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5388 ;*
5389 ; SET THE DUT C.R MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5390 ; AND WAIT UP C 5 SECONDS FOR THE MR BIT TO CLEAR.
5391 ;*
5392 027044 012701 011610 MOV 05000,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5393 027050 012702 000040 MOV 0BITC5,R2 ;WAITING FOR MASTER RESET BIT.
5394 027054 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5395 027056 016704 153160 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5396 027062 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5397 027064 004767 173254 JSR PC,SKPSTS ;SKIP THE SELFTEST.
5398 027070 004767 171070 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5399 027074 103012 BCC 4$ ;GO REPORT ERROR IF MR DID NOT CLEAR.
5400 ;*
5401 ; CHECK THAT THE RX DATA VALID BIT IS SET.
5402 ;*
5403 027076 012400 MOV (R4)+,R0 ;INCREMENT POINTER TO PNT TO DUT RBUF REG.
5404 027100 005714 TST (R4) ;TEST THE DUT RX.DATA.VALID BIT.
5405 027102 100016 BPL 6$ ;GO REPORT ERROR IF BIT IS NOT SET.
5406 ;*
5407 ; READ CHARACTERS FROM THE DUT RX FIFO AND WAIT FOR RX.DATA.VALID TO GO CLEAR.
5408 ;*
5409 027104 012705 001130 MOV 0600,R5 ;ALLOW READING 600 CHARS BEFORE ERROR.
5410 027110 011400 2$: MOV (R4),R0 ;READ A CHARACTER FROM THE RX FIFO.
5411 027112 100027 BPL 60$ ;EXIT TEST WITHOUT ERROR IF BIT IS CLEAR.
5412 027114 005305 DEC R5 ;COUNT THE CHARACTER JUST READ.
5413 027116 001374 BNE 2$ ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
5414 027120 000416 BR 8$ ;GO REPORT ERROR IF RX.DATA.VALID WOULDN'T CLR.
5415 ;*
5416 ; ERROR REPORTS:
5417 ;*
5418 ;*
5419 ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5420 027122 012767 001275 154640 4$: MOV 00701,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5421 027130 012701 005651 MOV 0EM0202,R1 ;SELECT ERROR MESSAGE.
5422 027134 104460 ERROR ;REPORT ERROR. >>>> ERROR 00701 <<<<
      027134 TRAP C$ERROR
5423 027136 000415 BR 60$ ;EXIT THE TEST.
5424

```

```

5425 ;REPORT THAT RX.DATA.VALID BIT WAS NOT SET AFTER A RESET COMPLETION.
5426 027140 012767 001276 154622 6$: MOV #0702.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5427 027146 012701 010245 MOV #EM0702,R1 ;SELECT ERROR MESSAGE.
5428 027152 ERROR ;REPORT ERROR. >>>>> ERROR #0702 <<<<<
      027152 104460 TRAP C$ERROR
5429 027154 000406 BR 60$ ;EXIT THE TEST.
5430
5431 ;REPORT THAT RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO.
5432 027156 012767 001277 154604 8$: MOV #0703.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5433 027164 012701 010425 MOV #EM0703,R1 ;SELECT ERROR MESSAGE.
5434 027170 ERROR ;REPORT ERROR. >>>>> ERROR #0703 <<<<<
      027170 104460 TRAP C$ERROR
5435
5436 027172 60$: SETPRI #PRI07 ;DISABLE ALL INTERRUPTS.
      027172 012700 000340 MOV #PRI07,R0
      027176 104441 TRAP C$SPRI
5437 027200 005067 153102 CLR CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
5438 027204 ENDTST
      027204 L10034:
      027204 104401 TRAP C$ETST

```



```

5440 .SBTTL  HARDWARE TEST - RLNA -
5441 ;+ *****
5442 ;+ - RBUF RX LINE NUMBER FIELD TEST -
5443 ;+ THIS TEST VERIFIES THAT THE DUT RBUF RX LINE NUMBER FIELD IS WORKING
5444 ;+ CORRECTLY BY UTILIZING THE SELFTEST CODES WHICH ARE PUT IN THE RX
5445 ;+ FIFO AFTER A BOARD RESET.
5446 ;+ *****
5447 ;-- *****
5448 027206 BGNTEST
5449 027206 SETPRI  #PRI05 ;ALLOW LTC INTERRUPTS. T8::
5450 027206 012700 000240 MOV  #PRI05,R0
5451 027212 104441 TRAP  C$SPRI
5452 027214 000010 TNUM  = TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5453 027222 012767 000010 153102 MOV  #TNUM,ISTNUM ;SET UP THE TEST NUMBER. (8)
5454 027230 012767 177777 153056 MOV  #-1,CIRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5455 027236 012767 000001 154530 MOV  #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5456 027236 012767 010610 154526 MOV  #EM0801,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5457 ;+
5458 ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5459 ; AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5460 027244 012701 011610 MOV  #5000.,R1 ;TIME OUT VALUE IS 5.0 SECONDS.
5461 027250 012702 000040 MOV  #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5462 027254 005003 CLR  R3 ;WAITING FOR BIT TO CLEAR.
5463 027256 016704 152760 MOV  CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5464 027262 010214 MOV  R2,(R4) ;SET THE DUT MASTER RESET BIT.
5465 027264 004767 173054 JSR  PC,SKPSTS ;SKIP THE SELFTEST.
5466 027270 004767 170670 JSR  PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5467 027274 103016 BCC  4$ ;GO REPORT ERROR IF MR DID NOT CLEAR.
5468 ;+
5469 ; READ CHARACTERS FROM THE DUT RX FIFO AND VERIFY THAT THE LINE NUMBERS ARE
5470 ; CORRECT.
5471 ; EIGHT CHARACTERS ARE READ FROM THE FIFO.
5472 027276 005001 CLR  R1 ;CLEAR THE LINE COUNTER.
5473 027300 012400 MOV  (R4)+,R0 ;INCREMENT POINTER TO PNT TO THE DUT RBUF REG.
5474 027302 011402 2$: MOV  (R4),R2 ;READ A CHARACTER FROM THE DUT RX FIFO.
5475 027304 010203 MOV  R2,R3
5476 027306 000303 SWAB R3
5477 027310 042703 177760 BIC  #177760,R3 ;REMOVE ALL BUT LINE NUMBER BITS.
5478 027314 020301 CMP  R3,R1 ;COMPARE WITH EXPECTED LINE NUMBER.
5479 027316 001017 BNE  6$ ;GO REPORT ERROR IF LINE NUMBERS DON'T MATCH.
5480 027320 005201 INC  R1 ;INCREMENT THE EXPECTED LINE NUMBER.
5481 027322 020127 000010 CMP  R1,#8. ;COMPARE NUMBER OF CODES READ WITH MAX.
5482 027326 001365 BNE  2$ ;LOOP UNTIL CODES FOR ALL LINES ARE READ.
5483 027330 000423 BR   60$ ;EXIT TEST WITHOUT ERROR.
5484 ;+
5485 ; ERROR REPORTS:
5486 ;+
5487 ; REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5488 4$: MOV  #0801.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5489 027332 012767 001441 154430 MOV  #ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5490 027340 012767 016136 154426 MOV  #EM0202,R1 ;SELECT ERROR MESSAGE.
5491 027346 012701 005651 ERROR ;REPORT ERROR.
5492 027352 104460 TRAP  C$ERROR

```

012

CZDHUBO DMU 11 FUNC TST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 107-1
HARDWARE TEST - RLNA

SEQ 144

```

5493 027354 000411          BR      604          ;EXIT THE TEST.
5494
5495                          ;REPORT THAT RX LINE NUMBER FIELD IS WRONG FOR SELFTEST CODE.
5496 027356 012767 001442 154404 64:      MOV      00802,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5497 027364 012767 016136 154402      MOV      0ER05C3,ERRBLK      ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5498 027372 012701 010656      MOV      0EM0802,R1      ;SELECT ERROR MESSAGE.
5499 027376      ERROR      ;REPORT ERROR.          >>>>> ERROR 00802 <<<<<
                    TRAP      C$ERROR
5500 027376 104460
5501 027400          604:      SETPRI      0PRI07          ;DISABLE ALL INTERRUPTS.
                    MOV      0PRI07,R0
                    TRAP      C$SPRI
                    027400 012700 000340
                    027404 104441
5502 027406 005067 152674      CLR      CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5503 027412      ENDTST
                    L10035:
                    027412 104401          TRAP      C$ETST

```



```

5505
5506
5507
5508
5509
5510
5511
5512
5513
5514
5515
5516
5517 027414
      027414
5518 027414
      027414 012700 000240
      027420 104441
5519      000011
5520 027422 012767 000011 152674
5521 027430 012767 177777 152650
5522 027436 012767 000001 154322
5523 027444 012767 001605 154316
5524 027452 012767 010726 154312
5525
5526
5527
5528
5529 027460 012701 005670
5530 027464 012702 000040
5531 027470 005003
5532 027472 016704 152544
5533 027476 004767 170462
5534 027502 103027
5535
5536
5537
5538 027504 010214
5539 027506 004767 172632
5540
5541
5542
5543
5544 027512 012704 000764
5545 027516 004767 170402
5546 027522 004767 171014
5547 027526 103015
5548
5549
5550
5551 027530 016702 152770
5552 027534 012703 002526
5553 027540 020203
5554 027542 001414
5555
5556
5557
5558

```

```

;SBTTL  HARDWARE TEST - BMPCHK
;*****
;      BMP CHECK TEST
;      THIS TEST IS USED TO VERIFY THAT THE DUT DOES NOT IMMEDIATELY FAIL
;      THE ON-BOARD BACKGROUND-MONITOR PROGRAM, AND HENCE INVALIDATE
;      SUCCEEDING TESTS.
;      THIS TEST LOOKS FOR BMP CODES IN THE FIFO FOR A SET PERIOD IMMEDIATELY
;      AFTER THE SELF-TEST IS SKIPPED.
;      ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE QUEUE AND ARE ALSO
;      REPORTED IN THIS TEST.
;*****
;      BGNTEST
;*****
;      SETPRI  @PRI05          ;ALLOW LTC INTERRUPTS.      T9:1
;                               MOV      @PRI05,R0
;                               TRAP     C1SPRI
;      TNUM ** TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
;      MOV      @TNUM,TSTNUM ;SET UP THE TEST NUMBER.      (9)
;      MOV      @1,CTRLCF    ;INDICATE THAT WE ARE WITHIN A TEST.
;      MOV      @1,ERRTYP    ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
;      MOV      @0901,ERRNBR ;SET THE ERROR NUMBER.
;      MOV      @EM0901,ERRMSG ;SET THE ERROR MESSAGE
;
;      WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
;      IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
;
;      MOV      @3000,R1      ;TIME-OUT VALUE IS 3.0 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      501          ;ABORT THE TEST IF MR DID NOT CLEAR.
;
;      RESET THE DUT, SKIP THE SELF-TEST.
;
;      MOV      R2,(R4)       ;SET THE DUT MASTER RESET BIT.
;      JSR      PC,SKPSTS     ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
;
;      WAIT FOR MASTER RESET TO CLEAR, DELAY FOR 500 MILLI-SECS BEFORE PURGING
;      THE FIFO.
;
;      MOV      @500,R4       ;TIME-OUT VALUE IS 500 MILLI SECONDS.
;      JSR      PC,DELAY      ;WAIT FOR BMP TO BEGIN EXECUTION.
;      JSR      PC,PURFIFO    ;PURGE THE FIFO, SAVING ANY BMP CODES.
;      BCC      501          ;ABORT THE TEST IF THE FIFO DID NOT CLEAR.
;
;      REPORT THE ERROR IF ANY BMP CODES WERE FOUND.
;
;      MOV      BMPCQB,R2     ;GET THE CONTENTS OF THE POINTER TO THE BMP Q.
;      MOV      @BMPCQB,R3    ;GET THE START ADDRESS OF THE QUEUE.
;      CMP      R2,R3         ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
;      BEQ      601          ;EXIT NO CODES IN THE QUEUE.
;
;      THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
;
;      ;REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN

```

D12

CZDHUBO DHU-11 FUNC 1ST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 108-1
HARDWARE TEST - RMPCHK

SEQ 146

```

5559 027544 012701 010766      MOV    #EM0902,R1      ;PASS THE MESSAGE TO BE REPORTED.
5560 027550      ERRDF    0901,EM0901,ER9301 ;      >>>>> ERROR #0901 <<<<<.
      027550      104455      TRAP    C$ERRDF
      027552      001605      .WORD    901
      027554      010726      .WORD    EM0901
      027556      017262      .WORD    ER9301
5561 027560      000405      BR      60$
5562
5563 027562      012767 001606 154200 50$;      MOV    #902,,ERRNBR      ;SET >>>>> ERROR #0902 <<<<<.
5564 027570      004767 172676      JSR     PC,TSABRT      ;REPORT NON-TEST RELATED ERROR.
5565
5566 027574      012700 000340      60$;      SETPRI  #PRI07      ;DISABLE ALL INTERRUPTS.
      027574      104441      MOV     #PRI07,R0
      027600      005067 152500      TRAP    C$SPRI
5567 027602      027606      CLR     CTRLCF      ;INDICATE THAT WE COMPLETED THE TEST.
5568 027606      ENDTST
      027606      104401      L10036:
      TRAP    C$ETST

```

```

5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580 027610
5581 027610
5582 027610
5583 027610
5584 027614
5585 012700 000240
5586 104441
5587 000012
5588 012767 000012 152500
5589 012767 177777 152454
5590 012767 000001 154126
5591 012767 011022 154124
5592 012767 016136 154120
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603 027654 012701 011610
5604 027660 012702 000040
5605 027664 005003
5606 027666 016704 152350
5607 027672 004767 170266
5608 027676 103037
5609
5610
5611
5612
5613
5614
5615
5616 027700 012701 000062
5617 027704 010214
5618 027706 004767 172432
5619 027712 004767 170246
5620 027716 103011
5621 027720 020127 000050
5622 027724 003015
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633
5634
5635
5636
5637
5638
5639
5640
5641
5642
5643
5644
5645
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102
6103
6104
6105
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191
6192
6193
6194
6195
6196
6197
6198
6199
6200
6201
6202
6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268
6269
6270
6271
6272
6273
6274
6275
6276
6277
6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
63
```

Address	Offset	Instruction	Comment
5624	027750	012701 011055	
5625	027754	MOV #EM1002,R1	;SELECT ERROR MESSAGE.
	027754	ERROR	;REPORT ERROR. >>>> ERROR #1001 <<<<
5626	027756	BR 60\$	;EXIT THE TEST. TRAP C\$ERROR
5627			
5628			;REPORT SKIP SELF-TEST COMPLETED TOO SOON.
5629	027760	012767 001752 154002 4\$:	MOV #1002,,ERRNBR ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5630	027766	012701 011142	MOV #EM1003,R1 ;SELECT ERROR MESSAGE.
5631	027772		ERROR ;REPORT ERROR. >>>> ERROR #1002 <<<<
	027772	104460	TRAP C\$ERROR
5632	027774	000405	BR 60\$;EXIT THE TEST.
5633			
5634	027776	012767 001753 153764 50\$:	MOV #1003,,ERRNBR ;SET ERROR NUMBER.
5635	030004	004767 172462	JSR PC,TSABRT ;REPORT NON-TEST RELATED ERROR.
5636			
5637	030010		60\$:
	030010	012700 000340	SETPRI #PRI07 ;DISABLE ALL INTERRUPTS.
	030014	104441	MOV #PRI07,RO
5638	030016	005067 152264	TRAP C\$SPRI
5639	030022		CLR CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
	030022		ENDTST
			L10037:
	030022	104401	TRAP C\$ETST

```

5641
5642
5643
5644
5645
5646
5647
5648
5649
5650 030024
      030024
5651 030024
      030024 012700 000240
      030030 104441
5652      000013
5653 030032 012767 000013 152264
5654 030040 012767 177777 152240
5655 030046 012767 000001 153712
5656 030054 012767 011220 153710
5657 030062 012767 016136 153704
5658
5659
5660
5661
5662 030070 012701 011610
5663 030074 012702 000040
5664 030100 005003
5665 030102 016704 152134
5666 030106 004767 170052
5667 030112 103044
5668
5669
5670
5671 030114 010214
5672 030116 004767 172222
5673
5674
5675
5676
5677 030122 012701 000005
5678 030126 012702 020000
5679 030132 010203
5680 030134 016704 152102
5681 030140 004767 170020
5682 030144 103020
5683
5684
5685
5686
5687
5688
5689
5690 030146 012701 000017
5691 030152 005003
5692 030154 004767 170004
5693 030160 107012
5694 030162 010105

;SBTTL  HARDWARE TEST          - DFKST -
;*****
;*          - DIAGNOSTIC FAIL BIT, SKIP SELF TEST TEST -
;*          THIS TEST VERIFIES THAT THE DIAGNOSTIC FAIL BIT OF THE DUT, CORRECTLY
;*          CHANGES STATE AS THE ON-BOARDED SELFTEST IS SKIPPED.
;*
;-- *****
      BGNIST
      T11::
      SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
      MOV     #PRI05,R0
      TRAP    C$SPRI
      TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV     #TNUM,T$TNUM    ;SET UP THE TEST NUMBER. (11)
      MOV     #-1,C$TRLCF     ;INDICATE THAT WE ARE WITHIN A TEST.
      MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
      MOV     #EM1101,ERRMSG   ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
      MOV     #ER0503,ERRBLK   ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
;
; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
; IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
;--
      MOV     #5000.,R1        ;TIME-OUT VALUE IS 5.0 SECONDS.
      MOV     #BIT05,R2        ;WAITING FOR MASTER RESET BIT.
      CLR     R3               ;WAITING FOR BIT TO CLEAR.
      MOV     CSRA,R4          ;BIT IS IN THE DUT'S CSR.
      JSR     PC,MSLGET        ;WAIT FOR DUT CSR_MR BIT TO CLEAR.
      BCC     50$              ;ABORT THE TEST IF MR DID NOT CLEAR.
;
; RESET THE DUT, SKIP THE SELF-TEST.
;--
      MOV     R2,(R4)          ;SET THE DUT MASTER RESET BIT.
      JSR     PC,SKPSTS        ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
;
; SET TIME-OUT OF 5 MILLI SECONDS, WAIT FOR DIAG_FAIL BIT TO SET.
; IF TIME-OUT OCCURS GO REPORT THE ERROR.
;--
      MOV     #5,R1            ;TIME-OUT VALUE IS 5 MILLI-SECONDS.
      MOV     #BIT13,R2        ;WAITING FOR DIAGNOSTIC FAIL BIT.
      MOV     R2,R3            ;WAITING FOR BIT TO SET.
      MOV     CSRA,R4          ;BIT IS IN THE DUT'S CSR.
      JSR     PC,MSLGET        ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
      BCC     4$              ;IF DIAG_FAIL DID NOT SET, GO REPORT ERROR.
;
; SET TIME-OUT OF 15 MILLI-SECS, WAIT FOR DIAG_FAIL TO CLEAR.
; IF TIME-OUT OCCURS GO REPORT THE ERROR.
; VERIFY THE DIAG_FAIL BIT IS IN A STABLE STATE BEFORE CONTINUING. LOOP
; BACK IF THE STATE WAS TRANSITORY, USING THE REMAINDER OF THE 15 MS TIME-OUT.
;
      MOV     #15.,R1          ;TIME-OUT VALUE IS 15 MILLI SECONDS.
      CLR     R3               ;WAITING FOR BIT TO CLEAR.
      JSR     PC,MSLGET        ;WAIT FOR DUT_CSR_DF BIT TO CLEAR.
      BCC     4$              ;IF DIAG_FAIL DID NOT CLEAR, GO REPORT ERROR.
      MOV     R1,R5            ;SAVE THE REMAINING TIME-OUT VALUE.
2$:

```

```

5695 030164 012701 000001      MOV    #1,R1      ;SET TIME-OUT OF 1 MILLI-SECOND.
5696 030170 052703 02000C      BIS    #BIT13,R3    ;WAIT FOR BIT TO SET.
5697 030174 004767 167764      JSR    PC,MSLGET    ;DOUBLE CHECK TO ELIMINATE NOISE PROBLEMS.
5698 030200 103016              BCC    60$          ;EXIT IF DIAG_FAIL BIT STILL CLEAR.
5699 030202 010501      MOV    R5,R1      ;PASS THE REMAINING TIME-OUT VALUE.
5700 030204 000762      BR     2$          ;LOOP TO CHECK AGAIN.
5701
5702      ; ERROR REPORTS:
5703
5704      ;REPORT DIAGNOSTIC FAIL BIT BAD.
5705 030206 012767 002115 153554 4$:      MOV    #1101,ERRNBR    ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5706 030214 012701 011517      MOV    #EM1205,R1    ;SELECT ERROR MESSAGE.
5707 030220              ERROR          ;REPORT ERROR.      >>>>> ERROR #1101 <<<<<
5708 030222 000405              BR     60$          ;EXIT THE TEST.      TRAP    C$ERROR
5709
5710 030224 012767 002116 153536 50$:      MOV    #1102,ERRNBR    ;SET THE ERROR NUMBER FOR TSABRT RTN.
5711 030232 004767 172234      JSR    PC,TSABRT    ;REPORT NON-TEST RELATED ERROR.
5712
5713 030236              60$:      SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
5714 030244 005067 152036              CLR    CTRLCF          ;INDICATE THAT WE COMPLETED A TEST.
5715 030250              ENDTST
5715 030250 104401              L10040:      TRAP    C$ETST

```



```

5717 .SBTTL HARDWARE TEST - SELFTEST -
5718 ;* *****
5719 ;* - SELF-TEST TEST -
5720 ;* THIS TEST VERIFIES THAT THE DUT'S SELF-TEST EXECUTES WITHIN THE
5721 ;* TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS
5722 ;* COMPLETION.
5723 ;*
5724 ;* *****
5725 BGNTST
5726 030252 SETPRI $PRI05 ;ALLOW LTC INTERRUPTS. T12::
      030252 MOV $PRI05,R0
      030252 TRAP C$SPRI
      030256 104441
5727 000014 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5728 030260 012767 000014 152036 MOV $TNUM,TSTNUM ;SET UP THE TEST NUMBER. (12)
5729 030266 012767 177777 152012 MOV $-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5730 030274 012767 000001 153464 MOV $1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5731 030302 012767 011273 153462 MOV $EM1201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5732 030310 012767 016136 153456 MOV $ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5733 ;*
5734 ;* WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5735 ;* IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5736 ;*
5737 030310 012701 011610 MOV $5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5738 030322 012702 000040 MOV $BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5739 030326 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5740 030330 016704 151706 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5741 030334 004767 167624 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5742 030340 103067 BCC 50$ ;ABORT THE TEST IF MR DID NOT CLEAR.
5743 ;*
5744 ;* DETERMINE IF THE SELF-TEST TAKES TOO SHORT OR TOO LONG A TIME TO COMPLETE.
5745 ;* SET-UP A TIME-OUT OF 5 SECONDS. IF MR IS CLEAR IN LESS THAN 1/2 SECOND, OR
5746 ;* GREATER THAN 5 SECONDS, REPORT THE ERROR.
5747 ;*
5748 030342 012701 011610 MOV $5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5749 030346 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5750 030350 004767 167610 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5751 030354 103034 BCC 4$ ;GO REPORT ERROR SELFTEST TOOK TOO LONG.
5752 030356 012702 011610 MOV $5000.,R2
5753 030362 160102 SUB R1,R2 ;CALCULATE # OF MS SELFTEST TO COMPLETE.
5754 030364 020227 000062 CMP R2,$50.
5755 030370 002435 BLT 6$ ;SELFTEST SKIPPED? YES, GO REPORT ERROR.
5756 030372 020227 000764 CMP R2,$500.
5757 030376 002441 BLT 8$ ;GO REP ERR IF SELFTEST COMPLETED IN < 1/2 SEC.
5758 ;*
5759 ;* SELF-TEST COMPLETED WITHIN 1SEC TO 5 SECONDS.
5760 ;* CHECK THE STATE OF THE DIAGNOSTIC FAIL BIT, REPORT ERROR IF IT IS SET.
5761 ;*
5762 030400 032714 020000 BIT $BIT13,(R4) ;DETERMINE IF THE DIAG_FAIL BIT IS CLEAR.
5763 030404 001412 BEQ 2$ ;SKIP ERROR REPORT IF BIT IS CLEAR.
5764 ;REPORT DIAGNOSTIC FAIL BIT BAD.
5765 030406 012767 002264 153354 MOV $1204.,ERRNBR ;SET ERROR NUMBER TO IN ERROR TABLE.
5766 030414 012701 011517 MOV $EM1205,R1 ;SELECT THE ERROR MESSAGE.
5767 030420 ERROR ;
      030420 104460 TRAP C$ERROR
5768 ;*
5769 ;* EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

```

```

5770
5771 030422 032767 000100 151574 BIT 0BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
5772 030430 001440 BEQ 60$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5773 ;DURING THE SOFTWARE QUESTIONS.
5774
5775 ;*
5776 ; VERIFY THAT THE SELF-TEST CODES IN THE FIFO ARE "GOOD" CODES ,IE THE DUT
5777 ; SUCCESSFULLY COMPLETED THE SELF-TEST.
5778 ; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>> 1205 THRU 1209 <<<<.
5779 030432 012767 002265 153330 2$: MOV 01205,ERRNBR ;SET ERROR NUMBER TO 1205.
5780 030440 004767 171050 JSR PC,RSTRPT ;CHECK SELF-TEST CODES IN THE FIFO.
5781 030444 000432 BR 60$ ;EXIT TEST.
5782
5783 ;*
5784 ; ERROR REPORTS:
5785 ;*
5786 030446 012767 002261 153314 4$: ;REPORT SELF-TEST TOOK TOO LONG TO COMPLETE.
5787 030454 012701 011321 MOV 01201,ERRNBR ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5788 030460 104460 MOV 0EM1202,R1 ;SELECT ERROR MESSAGE.
5789 030462 000423 ERROR ;REPORT ERROR. >>>> ERROR 01201 <<<<
5790 TRAP C$ERROR
5791 BR 60$ ;EXIT THE TEST.
5792
5793 ;REPORT SELF-TEST DID NOT EXECUTE AFTER DUT RESET.
5794 030464 012767 002262 153276 6$: MOV 01202,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5795 030472 012701 011463 MOV 0EM1204,R1 ;SELECT ERROR MESSAGE.
5796 030476 104460 ERROR ;REPORT ERROR. >>>> ERROR 01202 <<<<
5797 TRAP C$ERROR
5798 BR 60$ ;EXIT THE TEST.
5799
5800 ;REPORT SELF-TEST COMPLETED TOO SOON.
5801 030502 012767 002263 153260 8$: MOV 01203,ERRNBR ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5802 030510 012701 011405 MOV 0EM1203,R1 ;SELECT ERROR MESSAGE.
5803 030514 104460 ERROR ;REPORT ERROR. >>>> ERROR 01203 <<<<
5804 TRAP C$ERROR
5805 BR 60$ ;EXIT THE TEST.
5806
5807 030520 012767 002272 153242 50$: MOV 01210,ERRNBR ;SET THE ERROR NUMBER FOR TSABRT RTN.
5808 030526 004767 171740 JSR PC,TSABRT ;REPORT NON-TEST RELATED ERROR.
5809
5810 60$: SETPRI 0PRI07 ;DISABLE ALL INTERRUPTS.
5811 MOV 0PRI07,R0
5812 TRAP C$SPRI
5813
5814 CLR CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
5815 ENDTST
5816 L10041:
5817 TRAP C$SETST
5818 030544 104401

```



```

5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820 030546
5821 030546
5822 030546
5823 030554
5824 030562
5825 030570
5826 030576
5827 030604
5828 030612
5829
5830
5831
5832
5833 030620
5834 030624
5835 030630
5836 030632
5837 030636
5838 030642
5839
5840
5841
5842
5843
5844 030644
5845 030652
5846 030656
5847 030662
5848
5849
5850
5851
5852 030670
5853 030674
5854 030700
5855 030704
5856 030706
5857 030712
5858 030716
5859
5860
5861
5862
5863 030720

          .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 TBUFCT REGISTER ON CHANNEL 0.
          ;--
          MOV     @BIT05,@CSRA     ;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,@TXBFCA ;WRITE THE FAIL SELF-TEST CODE TO TBUFCT 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.

```

Address	Op Code	Op 1	Op 2	Op 3	Op 4	Op 5	Op 6	Op 7	Op 8	Op 9	Op 10	Op 11	Op 12	Op 13	Op 14	Op 15	Op 16	Op 17	Op 18	Op 19	Op 20	Op 21	Op 22	Op 23	Op 24	Op 25	Op 26	Op 27	Op 28	Op 29	Op 30	Op 31	Op 32	Op 33	Op 34	Op 35	Op 36	Op 37	Op 38	Op 39	Op 40	Op 41	Op 42	Op 43	Op 44	Op 45	Op 46	Op 47	Op 48	Op 49	Op 50	Op 51	Op 52	Op 53	Op 54	Op 55	Op 56	Op 57	Op 58	Op 59	Op 60	Op 61	Op 62	Op 63	Op 64	Op 65	Op 66	Op 67	Op 68	Op 69	Op 70	Op 71	Op 72	Op 73	Op 74	Op 75	Op 76	Op 77	Op 78	Op 79	Op 80	Op 81	Op 82	Op 83	Op 84	Op 85	Op 86	Op 87	Op 88	Op 89	Op 90	Op 91	Op 92	Op 93	Op 94	Op 95	Op 96	Op 97	Op 98	Op 99	Op 100	Op 101	Op 102	Op 103	Op 104	Op 105	Op 106	Op 107	Op 108	Op 109	Op 110	Op 111	Op 112	Op 113	Op 114	Op 115	Op 116	Op 117	Op 118	Op 119	Op 120	Op 121	Op 122	Op 123	Op 124	Op 125	Op 126	Op 127	Op 128	Op 129	Op 130	Op 131	Op 132	Op 133	Op 134	Op 135	Op 136	Op 137	Op 138	Op 139	Op 140	Op 141	Op 142	Op 143	Op 144	Op 145	Op 146	Op 147	Op 148	Op 149	Op 150	Op 151	Op 152	Op 153	Op 154	Op 155	Op 156	Op 157	Op 158	Op 159	Op 160	Op 161	Op 162	Op 163	Op 164	Op 165	Op 166	Op 167	Op 168	Op 169	Op 170	Op 171	Op 172	Op 173	Op 174	Op 175	Op 176	Op 177	Op 178	Op 179	Op 180	Op 181	Op 182	Op 183	Op 184	Op 185	Op 186	Op 187	Op 188	Op 189	Op 190	Op 191	Op 192	Op 193	Op 194	Op 195	Op 196	Op 197	Op 198	Op 199	Op 200	Op 201	Op 202	Op 203	Op 204	Op 205	Op 206	Op 207	Op 208	Op 209	Op 210	Op 211	Op 212	Op 213	Op 214	Op 215	Op 216	Op 217	Op 218	Op 219	Op 220	Op 221	Op 222	Op 223	Op 224	Op 225	Op 226	Op 227	Op 228	Op 229	Op 230	Op 231	Op 232	Op 233	Op 234	Op 235	Op 236	Op 237	Op 238	Op 239	Op 240	Op 241	Op 242	Op 243	Op 244	Op 245	Op 246	Op 247	Op 248	Op 249	Op 250	Op 251	Op 252	Op 253	Op 254	Op 255	Op 256	Op 257	Op 258	Op 259	Op 260	Op 261	Op 262	Op 263	Op 264	Op 265	Op 266	Op 267	Op 268	Op 269	Op 270	Op 271	Op 272	Op 273	Op 274	Op 275	Op 276	Op 277	Op 278	Op 279	Op 280	Op 281	Op 282	Op 283	Op 284	Op 285	Op 286	Op 287	Op 288	Op 289	Op 290	Op 291	Op 292	Op 293	Op 294	Op 295	Op 296	Op 297	Op 298	Op 299	Op 300	Op 301	Op 302	Op 303	Op 304	Op 305	Op 306	Op 307	Op 308	Op 309	Op 310	Op 311	Op 312	Op 313	Op 314	Op 315	Op 316	Op 317	Op 318	Op 319	Op 320	Op 321	Op 322	Op 323	Op 324	Op 325	Op 326	Op 327	Op 328	Op 329	Op 330	Op 331	Op 332	Op 333	Op 334	Op 335	Op 336	Op 337	Op 338	Op 339	Op 340	Op 341	Op 342	Op 343	Op 344	Op 345	Op 346	Op 347	Op 348	Op 349	Op 350	Op 351	Op 352	Op 353	Op 354	Op 355	Op 356	Op 357	Op 358	Op 359	Op 360	Op 361	Op 362	Op 363	Op 364	Op 365	Op 366	Op 367	Op 368	Op 369	Op 370	Op 371	Op 372	Op 373	Op 374	Op 375	Op 376	Op 377	Op 378	Op 379	Op 380</
---------	---------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	----------

5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907 031034
031034
5908 031034
031034 012700 000240
031040 104441
5909 000016
5910 031042 012767 000016 151254
5911 031050 012767 177777 151230
5912 031056 012767 000001 152702
5913 031064 012767 011635 152700
5914 031072 012767 016136 152674
5915
5916
5917
5918
5919 031100 012701 005670
5920 031104 012702 000040
5921 031110 005003
5922 031112 016704 151124
5923 031116 004767 167042
5924 031122 103131
5925
5926
5927
5928 031124 010214
5929 031126 004767 171212
5930 031132 012701 011610
5931 031136 004767 167022
5932 031142 103121
5933
5934
5935
5936
5937
5938
5939
5940 031144 012705 000040
5941 031150 012703 000143
5942 031154 010304
5943 031156 012767 002571 152604
5944 031164 012701 011674
5945
5946 031170 017702 151050
5947 031174 100077
5948
5949

```
.SBTTL  HARDWARE TEST          - ROMVER -
; * *****
; *          - ROM VERSION TEST -
; * THIS TEST VERIFIES THAT THE DUT'S SELF-TEST PLACES VALID ROM VERSION
; * NUMBERS IN THE FIFO AFTER IT HAS BEEN SKIPPED.  THE ROM VERSION NUMBERS
; * WILL BE REPORTED (ON THE FIRST PASS ONLY), IF AN AFFIRMATIVE ANSWER
; * WAS GIVEN TO THE SOFTWARE P-TABLE QUESTION.
; * *****
; - *****
      BGNTST
; *
; *          T14::
      SETPRI 0PRI05          ;ALLOW LTC INTERRUPTS.
; *
; *          MOV 0PRI05,R0
; *          TRAP C0SPRI
; *
; *          TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
; *          MOV 0TNUM,TSTNUM      ;SET UP THE TEST NUMBER. (14)
; *          MOV 0-1,CTRLCF        ;INDICATE THAT WE ARE WITHIN A TEST.
; *          MOV 01,ERRTYP        ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
; *          MOV 0EM1401,ERRMSG    ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
; *          MOV 0ER0503,ERRBLK    ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
; *
; *          ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
; *          ; IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
; *
; *          MOV 03000.,R1          ;TIME-OUT VALUE IS 3.0 SECONDS.
; *          MOV 0BIT05,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$                ;ABORT THE TEST IF MR DID NOT CLEAR.
; *
; *          ; SET THE MASTER RESET BIT, AND SKIP THE SELF TEST.
; *
; *          MOV R2,(R4)            ;SET THE MASTER RESET BIT.
; *          JSR PC,SKPSTS          ;SKIP THE SELF TEST.
; *          MOV 05000.,R1          ;TIME-OUT VALUE IS 5.0 SECONDS.
; *          JSR PC,MSLGET          ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
; *          BCC 50$                ;ABORT THE TEST IF MR DID NOT CLEAR.
; *
; *          ; REMOVE CHARACTERS FROM THE FIFO UNTIL EITHER:
; *          ; (A) THE FIFO IS PURGED. GO REPORT THE ERROR.
; *          ; (B) THE MAXIMUM TRY COUNTER IS ZERO, GO REPORT THE ERROR.
; *          ; (C) PROC_1'S ROM VERSION NUMBER WAS FOUND BEFORE PROC_2'S, GO REPORT ERROR.
; *          ; (D) BOTH ROM VERSION NUMBERS HAVE BEEN FOUND.
; *
; *          MOV 032.,R5            ;SET MAXIMUM TRY COUNTER.
; *          MOV 099.,R3            ;SET AN INVALID ROM VERSION NUMBER FOR PROC 1.
; *          MOV R3,R4              ;SET AN INVALID ROM VERSION NUMBER FOR PROC 2.
; *          MOV 01401.,ERRNBR      ;SET THE ERROR NUMBER TO 1401.
; *          MOV 0EM1402,R1          ;SELECT MESSAGE TO BE REPORTED IF FIFO EMPTY.
; *
; *          2$: MOV 0RBUFA,R2        ;READ THE NEXT CHAR FROM THE FIFO.
; *          RPL 1$                 ;GO REPORT ERROR IF FIFO EMPTY.
; *
; *          ; CHECK IF THE READ DATA IS A BMP CODE.
```

```

5950
5951 031176 012700 000301      ;
5952 031202 040200      MOV    #301,R0      ;SET-UP A BIT MASK OF A BMP CODE.
5953 031204 001003      BIC    R2,R0      ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
5954 031206 004767 171064      BNE    4$      ;BRANCH IF NOT A BMP CODE.
5955 031212 000435      JSR    PC,SAVBMP    ;SAVE THE BMP CODE ON THE QUEUE.
5956
5957      ;
5958      ; CHECK IF THE READ DATA IS A SELF-TEST CODE.
5959 031214 012700 000201      4$:      MOV    #201,R0      ;SET-UP A BIT MASK OF A SELFTEST CODE.
5960 031220 040200      BIC    R2,R0      ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
5961 031222 001431      BEQ    8$      ;BRANCH IF IT IS A SELFTEST CODE.
5962
5963      ;
5964      ; THE READ DATA IS A ROM VERSION NUMBER, DETERMINE WHICH ONE IT IS.
5965      ;
5966 031224 032702 000002      BIT    #BIT1,R2      ;CHECK THE PROCESSOR NUMBER BIT IN THE CODE.
5967 031230 001407      BEQ    6$      ;BRANCH IF IT IS PROC_1 ROM VERSION NUMBER.
5968 031232 010204      MOV    R2,R4      ;SAVE PROC_2 ROM VERSION NUMBER.
5969 031234 042704 177603      BIC    #177603,R4      ;CLEAR ANY UNWANTED BITS.
5970 031240 000241      CLC      ;CLEAR THE CARRY BIT.
5971 031242 006004      ROR    R4      ;SHIFT THE CODES ALONG TO GET THE ROM
5972 031244 006004      ROR    R4      ; VERSION NUMBER IN THE LOW 5 BITS.
5973 031246 000417      BR     8$      ;
5974 031250 010203      6$:      MOV    R2,R3      ;SAVE PROC_1 ROM VERSION NUMBER.
5975 031252 042703 177603      BIC    #177603,R3      ;CLEAR ANY UNWANTED BITS.
5976 031256 000241      CLC      ;CLEAR THE CARRY BIT.
5977 031260 006003      ROR    R3      ;SHIFT THE CODE ALONG TO GET THE ROM
5978 031262 006003      ROR    R3      ; VERSION NUMBER IN THE LOW 5 BITS.
5979 031264 020427 000143      CMP    R4,#99.      ;CHECK IF WE HAVE RECEIVE PROC_2 ROM CODE.
5980 031270 001016      BNE    10$      ;GO REPORT BOTH ROM VERSION NUMBERS.
5981
5982      ;
5983      ; RECEIVED ROM VERSION NUMBERS OUT OF SEQUENCE.
5984      ; IE, PROC_1'S ROM VERSION NUMBER FOUND IN THE FIFO BEFORE PROC_2'S.
5985 031272 012701 011762      ;
5986 031276 012767 002572 152464      MOV    #EM1403,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5987 031304 000433      MOV    #1402,ERRNBR      ;SET THE ERROR NUMBER.
5988      BR     12$      ;GO REPORT ERROR.
5989
5989 031306 005305      8$:      DEC    R5      ;DECREMENT THE MAX TRY COUNTER.
5990 031310 001327      BNE    2$      ;LOOP TO GET THE NEXT CHAR FROM THE FIFO.
5991 031312 012701 012035      MOV    #EM1104,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5992 031316 012767 002573 152444      MOV    #1403,ERRNBR      ;SET THE ERROR NUMBER.
5993 031324 000423      BR     12$      ;GIVE UP, GO REPORT ERROR.
5994
5995      ;
5996      ; IF THIS IS THE FIRST PASS, AND SOFTWARE P-TABLE QUESTION WAS ANSWERED YES.
5997      ; THEN REPORT THE ROM VERSION NUMBERS TO THE OPERATOR.
5998 031326 032767 000001 150670      10$:      BIT    #BIT0,OPTION      ;CHECK ON THE STATE OF THE SOFTWARE SWITCH.
5999 031334 001431      BEQ    60$      ;EXIT IF NO ROM VERSION PRINTOUT WAS REQUESTED.
6000 031336 026727 150750 000001      CMP    PASCNT,#1      ;CHECK IF THIS IS THE FIRST PASS.
6001 031344 003025      BGT    60$      ;EXIT IF ROM VERS HAVE ALREADY BEEN REPORTED.
6002 031346      PRINTB #EF1401,R3,R4      ;PRINT THE ROM VERSION NUMBERS.
        031346 010446      MOV    R4,-(SP)
        031350 010346      MOV    R3,-(SP)
        031352 012746 004154      MOV    #EF1401,-(SP)
        031356 012746 000003      MOV    #3,-(SP)

```

CZDHUBO DHU 11 FUNC TEST PART1 MACRO M1200 15 MAR 84 09:15 PAGE 113-2
HARDWARE TEST ROMVER

GEQ 157

Address	Op Code	Op 1	Op 2	Op 3	Op 4	Op 5	Op 6	Op 7	Op 8	Op 9	Op 10	Op 11	Op 12	Op 13	Op 14	Op 15	Op 16	Op 17	Op 18	Op 19	Op 20	Op 21	Op 22	Op 23	Op 24	Op 25	Op 26	Op 27	Op 28	Op 29	Op 30	Op 31	Op 32	Op 33	Op 34	Op 35	Op 36	Op 37	Op 38	Op 39	Op 40	Op 41	Op 42	Op 43	Op 44	Op 45	Op 46	Op 47	Op 48	Op 49	Op 50	Op 51	Op 52	Op 53	Op 54	Op 55	Op 56	Op 57	Op 58	Op 59	Op 60	Op 61	Op 62	Op 63	Op 64	Op 65	Op 66	Op 67	Op 68	Op 69	Op 70	Op 71	Op 72	Op 73	Op 74	Op 75	Op 76	Op 77	Op 78	Op 79	Op 80	Op 81	Op 82	Op 83	Op 84	Op 85	Op 86	Op 87	Op 88	Op 89	Op 90	Op 91	Op 92	Op 93	Op 94	Op 95	Op 96	Op 97	Op 98	Op 99	Op 100	Op 101	Op 102	Op 103	Op 104	Op 105	Op 106	Op 107	Op 108	Op 109	Op 110	Op 111	Op 112	Op 113	Op 114	Op 115	Op 116	Op 117	Op 118	Op 119	Op 120	Op 121	Op 122	Op 123	Op 124	Op 125	Op 126	Op 127	Op 128	Op 129	Op 130	Op 131	Op 132	Op 133	Op 134	Op 135	Op 136	Op 137	Op 138	Op 139	Op 140	Op 141	Op 142	Op 143	Op 144	Op 145	Op 146	Op 147	Op 148	Op 149	Op 150	Op 151	Op 152	Op 153	Op 154	Op 155	Op 156	Op 157	Op 158	Op 159	Op 160	Op 161	Op 162	Op 163	Op 164	Op 165	Op 166	Op 167	Op 168	Op 169	Op 170	Op 171	Op 172	Op 173	Op 174	Op 175	Op 176	Op 177	Op 178	Op 179	Op 180	Op 181	Op 182	Op 183	Op 184	Op 185	Op 186	Op 187	Op 188	Op 189	Op 190	Op 191	Op 192	Op 193	Op 194	Op 195	Op 196	Op 197	Op 198	Op 199	Op 200	Op 201	Op 202	Op 203	Op 204	Op 205	Op 206	Op 207	Op 208	Op 209	Op 210	Op 211	Op 212	Op 213	Op 214	Op 215	Op 216	Op 217	Op 218	Op 219	Op 220	Op 221	Op 222	Op 223	Op 224	Op 225	Op 226	Op 227	Op 228	Op 229	Op 230	Op 231	Op 232	Op 233	Op 234	Op 235	Op 236	Op 237	Op 238	Op 239	Op 240	Op 241	Op 242	Op 243	Op 244	Op 245	Op 246	Op 247	Op 248	Op 249	Op 250	Op 251	Op 252	Op 253	Op 254	Op 255	Op 256	Op 257	Op 258	Op 259	Op 260	Op 261	Op 262	Op 263	Op 264	Op 265	Op 266	Op 267	Op 268	Op 269	Op 270	Op 271	Op 272	Op 273	Op 274	Op 275	Op 276	Op 277	Op 278	Op 279	Op 280	Op 281	Op 282	Op 283	Op 284	Op 285	Op 286	Op 287	Op 288	Op 289	Op 290	Op 291	Op 292	Op 293	Op 294	Op 295	Op 296	Op 297	Op 298	Op 299	Op 300	Op 301	Op 302	Op 303	Op 304	Op 305	Op 306	Op 307	Op 308	Op 309	Op 310	Op 311	Op 312	Op 313	Op 314	Op 315	Op 316	Op 317	Op 318	Op 319	Op 320	Op 321	Op 322	Op 323	Op 324	Op 325	Op 326	Op 327	Op 328	Op 329	Op 330	Op 331	Op 332	Op 333	Op 334	Op 335	Op 336	Op 337	Op 338	Op 339	Op 340	Op 341	Op 342	Op 343	Op 344	Op 345	Op 346	Op 347	Op 348	Op 349	Op 350	Op 351	Op 352	Op 353	Op 354	Op 355	Op 356	Op 357	Op 358	Op 359	Op 360	Op 361	Op 362	Op 363	Op 364	Op 365	Op 366	Op 367	Op 368	Op 369	Op 370	Op 371	Op 372	Op 373	Op 374	Op 375	Op 376	Op 377	Op 378	Op 379	Op 380</
---------	---------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	----------

```

6018 .SBTTL  HARDWARE TEST - CSR84 -
6019 .....
6020 CSR BIT 4 TEST -
6021 THIS TEST VERIFIES THAT WHEN THIS BIT IS SET (AT THE SAME TIME
6022 AS MASTER RESET) THE OUT REMAINS INACTIVE WITH THE MASTER RESET
6023 BIT SET, AND WHEN CSR BIT 4 IS SUBSEQUENTLY CLEARED, THE BOARD
6024 BECOMES ACTIVE AND REPORTS SIX SKIP SELFTEST CODES IN THE RXFIFO.
6025 ANY BMP CODES FOUND IN THE FIFO ARE SAVED TO BE REPORTED LATER.
6026 .....
6027 .....
6028 .....
6029 BGNTEST
6030 031434 012700 000240 SETPRI 0PRI05 ALLOW LTC INTERRUPTS. T15::
6031 031434 104441 MOV 0PRI05,R0
6032 031440 TRAP C:SPRI
6033 000017 INUM == INUM + 1 INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6034 031442 012767 000017 150654 MOV 0TNUM,TSTNUM SET UP THE TEST NUMBER.
6035 031450 012767 177777 150630 MOV 0 1,CTRLCF INDICATE THAT WE ARE WITHIN A TEST.
6036 031456 012767 000001 152302 MOV 01,ERRTYP SET ERROR TYPE AS FATAL IN ERROR TABLE.
6037 031464 012767 002735 152276 MOV 01501,ERRNBR SET THE ERROR NUMBER TO 1501.
6038 031472 012767 012155 152272 MOV 0EM1501,ERRMSG SET THE ERROR MESSAGE ADDR.
6039 031500 012767 016136 152266 MOV 0ERC503,ERRBLK SET THE ERROR REPORTING ROUTINE.
6040
6041 ; WAIT FOR THE MASTER RESET BIT TO CLEAR. REPORT THE ERROR IF IT FAILS
6042 ; TO CLEAR.
6043 ;
6044 031506 012701 011610 MOV 05000,R1 SET THE TIME-OUT VALUE OF 5 SECS.
6045 031512 012702 000040 MOV 0BIT05,R2 INDICATE TO TEST BIT 5.
6046 031516 005003 CLR R3 INDICATE TO TEST FOR BIT CLEAR.
6047 031520 016704 150516 MOV CSRA,R4 INDICATE TO TEST THE CSR REG.
6048 031524 004767 166550 JSR PC,MSLOOP WAIT FOR THE BIT TO CLEAR.
6049 031530 103115 BCC 501 JUMP TO REPORT ERROR IF BIT FAILED TO CLEAR.
6050
6051 ;
6052 ; SET THE MASTER RESET AND CSR BIT 4 BIT, AND THEN WAIT 5.5 SECS TO ENSURE
6053 ; THAT THE MR BIT DOESN'T CLEAR.
6054 ;
6055 031532 005267 152252 INC ERRNBR SET THE ERROR NUMBER TO 1502.
6056 031536 012777 000060 150476 MOV 060,0CSR4 RESET THE BOARD WITH BIT 4 SET.
6057
6058 ;
6059 ; VERIFY THAT CSR BIT 4 IS SET.
6060 031544 017700 150472 MOV 0CSR4,R0 READ THE CSR.
6061 031550 032700 000020 BIT 0BIT04,R0 TEST BIT 4.
6062 031554 001477 BEQ 401 EXIT WITH ERROR IF THE BIT IS CLEAR.
6063
6064 ;
6065 ; WAIT 5 SECONDS FOR THE MR BIT TO CLEAR.
6066 ;
6067 031556 005267 152206 INC ERRNBR SET THE ERROR NUMBER TO 1503.
6068 031562 012701 012574 MOV 05500,R1 SET THE TIME OUT VALUE OF 5.5 SECS.
6069 031566 004767 166506 JSR PC,MSLOOP WAIT FOR THE MASTER RESET BIT TO CLEAR.
6070 031572 103470 BCS 401 REPORT THE ERROR IF THE MR BIT CLEARED.
6071 ;

```



```

6072
6073
6074 031574 017705 150442
6075 031600 042705 000020
6076 031604 010577 150432
6077
6078
6079
6080 031610 005267 152154
6081 031614 017705 150422
6082 031620 032705 000020
6083 031624 001053
6084
6085
6086
6087 031626 005267 152136
6088 031632 012701 000764
6089 031636 004767 166436
6090 031642 103044
6091
6092
6093
6094
6095
6096 031644 012767 017126 152122
6097 031652 012701 015241
6098
6099 031656 012704 000006
6100
6101 031662 012767 002742 152100 23:
6102 031670 017702 150350
6103 031674 100033
6104 031676 010203
6105 031700 042703 177400
6106 031704 012705 000301
6107 031710 040305
6108 031712 001003
6109 031714 004757 170356
6110 031720 000760
6111
6112 031722 020327 000203
6113 031726 001407
6114
6115
6116
6117 031730 005267 152034
6118 031734
031734 104460
6119
6120
6121
6122
6123 031736 032767 000100 150260
6124 031744 001411
6125
6126 031746 001304
6127 031750 001344

; CLEAR CSR BIT 4 AND VERIFY THAT THE MASTER RESET BIT ALSO CLEARS.
;
MOV    BCSRA,R5      ; READ THE CSR.
BIC    0BIT04,R5     ; CLEAR BIT 4.
MOV    R5,BCSRA      ; RESTORE THE CONTENTS OF THE CSR.
;
; VERIFY THAT CSR BIT 4 CLEARED.
;
INC    ERRNR         ; SET THE ERROR NUMBER TO 1504.
MOV    BCSRA,R5      ; READ THE CSR.
BIT    0BIT04,R5     ; TEST BIT 4.
BNE    40$           ; BRANCH AND REPORT ERROR IF SET.
;
; WAIT FOR THE MR BIT TO CLEAR.
;
INC    ERRNR         ; SET THE ERROR NUMBER TO 1505.
MOV    0500.,R1      ; SET A TIME-OUT OF 1/2 SECS.
JSR    PC,MSLOOP     ; WAIT FOR THE MR BIT TO CLEAR.
BCC    40$           ; JUMP AND REPORT ERROR.
; MR BIT FAILED TO CLEAR.
;
; READ SIX CHARACTERS FROM THE RXFIFO AND VERIFY THEY ARE SKIP-SELFTEST
; CODES.  SAVE ANY BMP CODES FOUND TO BE REPORTED LATER.
;
MOV    0ER9008,ERRBLK ; SET UP THE ERROR ROUTINE.
MOV    0EM9024,R1     ; SET THE ERROR MESSAGE.
; IMPROPER SELFTEST CODE FOUND.
MOV    06.,R4         ; SET THE NUMBER OF CHAR'S TO READ.
;
MOV    01506.,ERRNR   ; SET THE ERROR NUMBER TO 1506.
MOV    0RBUFA,R2      ; READ A CODE FROM THE RXFIFO.
BPL    50$           ; EXIT WITH ERROR IF THE FIFO IS EMPTY.
MOV    R2,R3          ; COPY THE CODE.
BIC    0177400,R3     ; CLEAR THE LINE NUMBER AND ERROR FLAGS.
MOV    0501,R5        ; SET THE BMP CODE MASK.
BIC    R3,R5          ; CHECK IF THE CODE IS A BMP CODE.
BNE    41$           ; AVOID SAVING THE BMP CODE IF IT ISN'T.
JSR    PC,SAVBMP      ; SAVE THE BMP CODE.
BR     23$           ; AVOID COUNTING THIS CODE, AND BRANCH
; TO READ MORE DATA FROM THE RXFIFO.
41$:    CMP    R3,0203 ; IS THE CODE A SKIP SELFTEST?
BEQ    61$           ; BRANCH TO AVOID THE ERROR IF IT IS.
;
; REPORT UNEXPECTED SELFTEST CODE FOUND.
;
INC    ERRNR         ; SET THE ERROR NUMBER TO 1507.
ERROR  ; REPORT THE ERROR.
;
; IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED THEN EXIT
; THE TEST WITH THE TEST FAILURE MESSAGE.
;
BIT    0BIT06,OPTION  ; HAS EXTENDED ERROR REPORTING BEEN REQUESTED?
BEQ    60$           ; EXIT THE TEST IF IT HASN'T.
;
61$:    DEC    R4      ; DECREMENT THE CODE COUNT.
BNE    23$           ; BRANCH AND READ ANOTHER CODE IF NOT ALL

```

TRAP C\$ERROR

E13

C:\DHUBO\DHU_11 FUNC TEST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 114-2

SEQ 160

```

6128
6129 031752 000406          BR      60$      ;HAVE BEEN READ.
6130                                     ;OTHERWISE, EXIT THE TEST, THE DUT HAS PASSED
6131                                     ;THIS TEST.
6132
6133          ;
6134          ; REPORT THE ERROR "CSR BIT 4 BAD" AND EXIT THIS TEST.
6135          ;
6136
6137 031754 012701 012203    40$:  MOV    0EM1502,R1      ;SET UP THE EXTENDED ERROR MESSAGE AS,
6138                                     ; "CSR BIT 4 BAD".
6139 031760 104460          ERROR      ;REPORT THE ERROR.
6140 031762 000402          BR      60$      ;EXIT THE TEST.          .RAP    C$ERROR
6141
6142          ;
6143          ; REPORT A NON-RELATED TEST ERROR.
6144 031764 004767 170502    50$:  JSR    PC,TSABRT      ;REPORT THE ERROR.
6145
6146 031770 005067 150312    60$:  CLR    CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6147
6148 031774          ENDTST
        031774
        031774 104401          L10044:
                                TRAP    C$ETST

```



```

6150 .SBTTL  HARDWARE TEST - REGWRW -
6151 ;* *****
6152 ;* - DEVICE REGISTER WORD ACCESS READ AND WRITE TEST -
6153 ;*
6154 ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
6155 ;* CORRECTLY USING WORD ACCESSES.
6156 ;*
6157 ;* *****
6158
6159 031776          BGNTST
6160 031776          SETPRI 0PRI05          ;ALLOW THE LTC TO INTERRUPT.
6161 031776          T16::
6162 032002          MOV 0PRI05,R0
6163 032002          TRAP C$SPRI
6164 032004          TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6165 032012          MOV 0TNUM,TSTNUM      ;SET UP THE TEST NUMBER. (16)
6166 032020          MOV 0-1,CTRLCF       ;INDICATE THAT WE ARE WITHIN A TEST.
6167 032026          MOV 01,ERRTYP        ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6168 032034          MOV 01601,ERRNBR     ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6169 032042          MOV 0EM1604,ERRMSG   ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6170 032046          CLR ERSNRF           ;CLEAR THE ERROR SUMMARY FLAGS.
6171 032052          MOV 0ERCNTB,R0
6172 032056          JSR PC,CLR16W        ;CLEAR THE ERROR COUNTER TABLE.
6173          CLR EXOERR                  ;CLEAR THE "EXIT ON ERROR" FLAG
6174
6175 ;*
6176 ;* RESET THE DUT TO A KNOWN STATE. DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6177 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6178 ;* THIS SUBROUTINE REPORTS ERRORS >>>> 1601 <<<<.
6179
6180 JSR PC,RESETT      ;RESET THE DHU-11. REPORT ANY ERRORS FOUND.
6181 RCS 0-6           ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
6182 JMP 60$          ;YES, EXIT THE TEST.
6183
6184 ;*
6185 ;* VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR
6186 ;*
6187 INC ERRNBR        ;SET THE ERROR REPORT NUMBER TO 1602.
6188 MOV 017,R2       ;SET LOOP COUNT.
6189 MOV CSRA,R4       ;GET CSR ADDRESS.
6190 2$: MOV R2,(R4)    ;WRITE COUNT TO CSR.
6191 MOV (R4),R1       ;READ BACK THE CONTENTS OF THE CSR
6192 BIC 0177760,R1    ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
6193 CMP R1,R2         ;CHECK FOR CORRECT DATA WRITTEN/READ.
6194 BEQ 4$            ;IS EXPECTED DATA BAD? NO, SKIP ERROR REPORT.
6195 ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE 0 (D)."
6196 MOV 0ER1601,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
6197 CLR R3             ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
6198 CLR R5            ;CAUSE REPORT OF LINE 0.
6199 ERROR            ; >>>> ERROR 0 1602 <<<<
6200 TRAP C$ERROR
6201
6202 ;*
6203 ;* EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED.
6204 ;*
6205 BIT 0BIT06,OPTION ;HAS EXTENDED ERROR BEEN REQUESTED ?
6206 BEQ 60$          ;EXIT THE TEST IF IT HASN'T.
6207
6208 4$: DEC R2        ;DECREMENT LOOP COUNT IND.ADD.REG ADDRESS.
6209 BGE 2$           ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.

```

```

6203
6204
6205
6206
6207
6208 032154 005267 151610
6209 032160 005003
6210 032162 012704 000002
6211 032166 004767 166704
6212
6213
6214
6215
6216 032172 005767 150106
6217 032176 001017
6218
6219
6220
6221
6222
6223 032200 012767 003106 151562
6224 032206 005003
6225 032210 005404
6226 032212 004767 166660
6227
6228
6229
6230
6231 032216 005767 150062
6232 032222 001005
6233
6234
6235
6236
6237 032224 012767 003111 151536
6238 032232 004767 167072
6239 032236 005067 150044
6240 032242
      032242 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     60$         ;EXIT IF IT IS.

;+
; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
; ACTIVE LINES. BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1606 - 1608 <<<<.
;-
      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     60$         ;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

60$:

```

```

L10045: TRAP C$ETST

```

```

6242 .SSTTL  HARDWARE TEST - REGWRM -
6243 ;* *****
6244 ;* - DEVICE REGISTER WORD ACCESS READ/MODIFY/WRITE TEST -
6245 ;*
6246 ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE WRITTEN CORRECTLY
6247 ;* USING WORD READ/MODIFY/WRITE ACCESSES.
6248 ;*
6249 ;* - *****
6250
6251 BGNTST
6252 032244 SETPRI 0PRI05 ;ALLOW THE LTC TO INTERRUPT.
6253 032244 T17::
6254 032244 012700 000240 MOV 0PRI05,R0
6255 032250 104441 TRAP C$SPRI
6256 032252 000021 TNUM ** TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6257 032252 012767 000021 150044 MOV 0TNUM,T$TNUM ;SET UP THE TEST NUMBER. (17)
6258 032260 012767 177777 150020 MOV 0-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
6259 032266 012767 000001 151472 MOV 01,ERRTYP ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6260 032274 012767 003245 151466 MOV 01701,ERRNBR ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6261 032302 012767 012360 151462 MOV 0EM1701,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6262 032310 005067 150146 CLR ERSMRF ;CLEAR THE ERROR SUMMARY FLAGS.
6263 032314 012700 002464 MOV 0ERCNTB,R0
6264 032320 004767 165500 JSR PC,CLR16W ;CLEAR THE ERROR COUNTER TABLE.
6265 032324 005067 147754 CLR EXOERR ;CLEAR THE "EXIT ON ERROR" FLAG
6266
6267 ;*
6268 ;* RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6269 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6270 ;* THIS SUBROUTINE REPORTS ERRORS >>>> 1701 <<<<.
6271 ;*
6272 JSR PC,RESETT ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6273 BCS 0+6 ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
6274 JMP 60$ ;YES, EXIT THE TEST.
6275
6276 ;*
6277 ;* THE READ/MODIFY/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR IS
6278 ;* NOT TESTED THIS THIS FORM OF ACCESS IS ILLEGAL.
6279 ;*
6280
6281 ;*
6282 ;* WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
6283 ;* ACTIVE LINES USING R/M/W. BEFORE WRITING EACH PATTERN, CLEAR ALL THE BITS.
6284 ;* REGST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1703 - 1705 <<<<.
6285 ;*
6286 MOV 01703,ERRNBR ;SET THE ERROR NUMBER TO 1703.
6287 CLR R3 ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6288 MOV 01,R4 ;INDICATE R/M/W ACCESS, CLEAR FIRST.
6289 JSR PC,REGST ;WRITE AND VERIFY DATA PATTERNS.
6290
6291 ;*
6292 ;* EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6293 ;* NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6294 ;*
6295 TST EXOERR ;IS THE "EXIT ON ERROR" FLAG SET ?
6296 BNE 60$ ;EXIT IF IT IS.
6297
6298 ;*
6299 ;* WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
6300 ;* ACTIVE LINES USING R/M/W. BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
6301 ;* REGST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1706 - 1708 <<<<.

```

```

6296
6297 032370 012767 003252 151372 ; MOV    #1706.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6298 032376 005003          CLR    R3      ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6299 032400 005404          NEG    R4      ;INDICATE R/M/W ACCESS, SET FIRST.
6300 032402 004767 166470    JSR    PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6301
6302 ;*
6303 ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6304 ; NOT BEEN REQUESTED. I.E. EXOERR IS NON-ZERO.
6305 032406 005767 147672    ;*
6306 032412 001005          TST    EXOERR    ;IS THE "EXIT ON ERROR" FLAG SET ?
6307          BNE    60$      ;EXIT IF IT IS.
6308
6309 ;*
6310 ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6311 ; THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>> ERROR # 1709 <<<<
6312 032414 012767 003255 151346 ;*
6313 032422 004767 166702    MOV    #1709.,ERRNBR ;SET UP ERROR NUMBER FOR NEXT RTN.
6314 032426 005067 147654    JSR    PC,REPSMR ;REPORT ERROR SUMMARY IF NECESSARY.
6315 032432          CLR    CTRLCF    ;INDICATE THAT WE COMPLETED THE TEST.
        032432 104401          ENDTST
                                L10046:
                                TRAP    C$ETST

```

```

6317 .SBTTL HARDWARE TEST - REGBRW -
6318 ;+ *****
6319 ;+ - DEVICE REGISTER BYTE ACCESS READ AND WRITE TEST -
6320 ;+
6321 ;+ THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
6322 ;+ CORRECTLY USING BYTE ACCESSES.
6323 ;+
6324 ;+ *****
6325
6326 032434 BGNIST
6327 032434 T18:;
032434 SETPRI 0PRI05 ;ALLOW THE LTC TO INTERRUPT.
032440 104441 MOV 0PRI05,R0
032440 000022 TRAP C$SPRI
6328 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6329 032442 012767 000022 147654 MOV 0TNUM,TSTNUM ;SET UP THE TEST NUMBER. (18)
6330 032450 012767 177777 147630 MOV 0-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
6331 032456 012767 000001 151302 MOV 01,ERRTYP ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6332 032464 012767 003411 151276 MOV 01801,ERRNBR ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6333 032472 012767 012443 151272 MOV 0EM1801,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6334 032500 005067 147756 CLR ERSRFR ;CLEAR THE ERROR SUMMARY FLAGS.
6335 032504 012700 002464 MOV 0ERCNTB,R0
6336 032510 004767 165310 JSR PC,CLR16W ;CLEAR THE ERROR COUNTER TABLE.
6337 032514 005067 147564 CLR EXOERR ;CLEAR THE "EXIT ON ERROR" FLAG.
6338 ;+
6339 ;+ RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6340 ;+ CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6341 ;+ THIS SUBROUTINE REPORTS ERRORS >>>> 1801 <<<<.
6342 ;+
6343 032520 004767 166632 JSR PC,RESETT ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6344 032524 103402 BCS 0+6 ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
6345 032526 000167 000212 JMP 60$ ;YES, EXIT THE TEST.
6346 032532 012767 003412 151230 MOV 01802,ERRNBR ;SET THE ERROR REPORT NUMBER TO 1802.
6347 ;+
6348 ;+ VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR.
6349 ;+ US BYTE ACCESSES.
6350 ;+
6351 032540 012702 000017 MOV 017,R2 ;SET LOOP COUNT.
6352 032544 016704 147472 MOV CSRA,R4 ;GET CSR ADDRESS.
6353 032550 110214 2$: MOV R2,(R4) ;WRITE COUNT TO CSR.
6354 032552 111401 MOV R4,R1 ;READ BACK THE CONTENTS OF THE CSR
6355 032554 042701 177760 BIC 0177760,R1 ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
6356 032560 020102 CMP R1,R2 ;CHECK FOR CORRECT DATA WRITTEN/READ.
6357 032562 001412 BEQ 4$ ;IS EXPECTED DATA BAD? NO, SKIP ERROR REPORT.
6358 ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE 0 (D)."
6359 032564 012767 016416 151202 MOV 0ER1601,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
6360 032572 005003 CLR R3 ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
6361 032574 005005 CLR R5 ;CAUSE REPORT OF LINE 0.
6362 032576 104460 ERROR ; >>>> ERROR 0 1802 <<<< TRAP C$ERROR
6363 ;+
6364 ;+ EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
6365 ;+
6366 032600 032767 000100 147416 BIT 0BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
6367 032606 001456 BEQ 60$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
6368 ;+
6369 ;+ DURING THE SOFTWARE QUESTIONS.

```

```

6370 032610 005302 4$: DEC R2 ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
6371 032612 002356 BGE 2$ ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
6372
6373 ;*
6374 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6375 ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6376 ; EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6377 ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> ERROR 1803 - 1805 <<<<<.
6378 032614 005267 151150 INC ERRNBR ;SET THE ERROR NUMBER TO 1803.
6379 032620 012703 177777 MOV 0-1,R3 ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6380 032624 012704 000002 MOV 02,R4 ;INDICATE R/W ACCESS. CLEAR FIRST.
6381 032630 004767 166242 JSR PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6382
6383 ;*
6384 ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6385 ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6386 032634 005767 147444 TST EXOERR ;IS THE "EXIT ON ERROR" FLAG SET ?
6387 032640 001041 BNE 60$ ;EXIT IF IT IS.
6388
6389 ;*
6390 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6391 ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6392 ; EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6393 ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> ERROR 1806 - 1808 <<<<<.
6394
6395 032642 012767 003416 151120 MOV 01806.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6396 032650 005403 NEG R3 ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6397 032652 004767 166220 JSR PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6398
6399 ;*
6400 ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6401 ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6402 032656 005767 147422 TST EXOERR ;IS THE "EXIT ON ERROR" FLAG SET ?
6403 032662 001030 BNE 60$ ;EXIT IF IT IS.
6404
6405 ;*
6406 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6407 ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6408 ; EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6409 ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> ERROR 1809 - 1811 <<<<<.
6410 032664 012767 003421 151076 MOV 01809.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6411 032672 005403 NEG R3 ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6412 032674 005404 NEG R4 ;INDICATE R/W ACCESS. SET FIRST
6413 032676 004767 166174 JSR PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6414
6415 ;*
6416 ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6417 ; NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6418 032702 005767 147376 TST EXOERR ;IS THE "EXIT ON ERROR" FLAG SET ?
6419 032706 001016 BNE 60$ ;EXIT IF IT IS.
6420
6421 ;*
6422 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6423 ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6424 ; EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6425 ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> ERROR 1812 - 1814 <<<<<.
6426 032710 012767 003424 151052 MOV 01812.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.

```



```

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

```

```

L10047: TRAP    C$ETST

```

```

6444 .SBTTL  HARDWARE TEST - REGBRM -
6445 ;* *****
6446 ;* - DEVICE REGISTER BYTE ACCESS READ/MODIFY/WRITE TEST -
6447 ;*
6448 ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
6449 ;* CORRECTLY USING BYTE ACCESSES IN READ/MODIFY/WRITE MODE.
6450 ;*
6451 ;* *****
6452 ;*
6453 032752 BGNSTST
6454 032752 T19::
032752 SETPRI #PRI05 ;ALLOW THE LTC TO INTERRUPT.
032756 104441 MOV #PRI05,RO
032756 000023 TRAP C$SPRI
6455 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6456 032760 012767 000023 147336 MOV #TNUM,ISTNUM ;SET UP THE TEST NUMBER. (19)
6457 032766 012767 177777 147312 MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
6458 032774 012767 000001 150764 MOV #1,ERRTYP ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6459 033002 012767 003555 150760 MOV #1901,,ERRNBR ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6460 033010 012767 012517 150754 MOV #EM1901,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6461 033016 005067 147440 CLR ERSRFR ;CLEAR THE ERROR SUMMARY FLAGS.
6462 033022 012700 002464 MOV #ERCNTB,RO
6463 033026 004767 164772 JSR PC,CLR16W ;CLEAR THE ERROR COUNTER TABLE.
6464 033032 005067 147246 CLR EXOERR ;CLEAR THE "EXIT ON ERROR" FLAG.
6465 ;*
6466 ;* RESET THE DUT TO A KNOWN STATE. DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6467 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6468 ;* THIS SUBROUTINE REPORTS ERRORS >>>> 1901 <<<<.
6469 ;*
6470 033036 004767 166314 JSR PC,RESET ;RESET THE DHU-11. REPORT ANY ERRORS FOUND.
6471 033042 103402 BCS .+6 ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
6472 033044 000167 000136 JMP 60$ ;YES, EXIT THE TEST.
6473 033050 012767 003557 150712 MOV #1903,,ERRNBR ;SET THE ERROR REPORT NUMBER TO 1903.
6474 ;*
6475 ;* THE READ/MODIFY/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR IS NOT
6476 ;* TESTED SINCE THIS IS AN ILLEGAL FORM OF ACCESS TO THIS REGISTER.
6477 ;*
6478 ;*
6479 ;*
6480 ;*
6481 ;* WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6482 ;* REGISTERS ON ALL ACTIVE LINES. USE READ/MODIFY/WRITE ACCESSES. BEFORE
6483 ;* WRITING EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6484 ;* REGSTR ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1903 - 1905 <<<<.
6485 ;*
6486 033056 005267 150706 INC ERRNBR ;SET THE ERROR NUMBER TO 1903.
6487 033062 012703 177777 MOV #1,R3 ;INDICATE THAT LC BYTE ACCESSES ARE TO BE USED.
6488 033066 012704 000001 MOV #1,R4 ;INDICATE R/M/W ACCESS, CLEAR FIRST.
6489 033072 004767 166000 JSR PC,REGSTR ;WRITE AND VERIFY DATA PATTERNS.
6490 ;*
6491 ;* EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6492 ;* NOT BEEN REQUESTED, I.E. EXOERR IS NON-ZERO.
6493 ;*
6494 033076 005767 147202 IST EXOERR ;IS THE "EXIT ON ERROR" FLAG SET ?
6495 033102 001041 BNE 60$ ;EXIT IF IT IS.
6496 ;*
6497 ;* WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL

```



```

6498      ; REGISTERS ON ALL ACTIVE LINES.  USE READ/MODIFY/WRITE ACCESSES.  BEFORE
6499      ; WRITING EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6500      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1906 - 1908 <<<<.
6501      ;
6502 033104 012767 003562 150656      MOV    #1906.,ERRNBR    ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6503 033112 005403                    NEG    R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6504 033114 004767 165756            JSR    PC,REGTST        ;WRITE AND VERIFY DATA PATTERNS.
6505      ;
6506      ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6507      ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
6508      ;
6509 033120 005767 147160            TST    EXOERR            ;IS THE "EXIT ON ERROR" FLAG SET ?
6510 033124 001030                    BNE    60$              ;EXIT IF IT IS.
6511      ;
6512      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6513      ; REGISTERS ON ALL ACTIVE LINES.  USE READ/MODIFY/WRITE ACCESSES.  BEFORE
6514      ; WRITING EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6515      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1909 - 1911 <<<<.
6516      ;
6517 033126 012767 003565 150634      MOV    #1909.,ERRNBR    ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6518 033134 005403                    NEG    R3              ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6519 033136 005404                    NEG    R4              ;INDICATE R/M/W ACCESS, SET FIRST.
6520 033140 004767 165732            JSR    PC,REGTST        ;WRITE AND VERIFY DATA PATTERNS.
6521      ;
6522      ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6523      ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
6524      ;
6525 033144 005767 147134            TST    EXOERR            ;IS THE "EXIT ON ERROR" FLAG SET ?
6526 033150 001016                    BNE    60$              ;EXIT IF IT IS.
6527      ;
6528      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6529      ; REGISTERS ON ALL ACTIVE LINES.  USE READ/MODIFY/WRITE ACCESSES.  BEFORE
6530      ; WRITING EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6531      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1912 - 1914 <<<<.
6532      ;
6533 033152 012767 003570 150610      MOV    #1912.,ERRNBR    ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6534 033160 005403                    NEG    R3              ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6535 033162 004767 165710            JSR    PC,REGTST        ;WRITE AND VERIFY DATA PATTERNS.
6536      ;
6537      ; EXIT THE TEST IF AN ERROR HAS BEEN FOUND AND EXTENDED ERROR REPORTING HAS
6538      ; NOT BEEN REQUESTED, I.E.  EXOERR IS NON-ZERO.
6539      ;
6540 033166 005767 147112            TST    EXOERR            ;IS THE "EXIT ON ERROR" FLAG SET ?
6541 033172 001005                    BNE    60$              ;EXIT IF IT IS.
6542      ;
6543      ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6544      ; THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>> ERROR # 1915 <<<<
6545      ;
6546 033174 012767 003573 150566      MOV    #1915.,ERRNBR    ;SET UP ERROR NUMBER FOR NEXT RTN.
6547 033202 004767 166122            JSR    PC,REPSMR        ;REPORT ERROR SUMMARY IF NECESSARY.
6548 033206 005067 147074            CLR    CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
6549 033212                                ENDTST
        033212 104401

```

L10050: TRAP C\$ETST

```

6551 .SRTT1 HARDWARE TEST IDBIT
6552 .....
6553 .....
6554 .....
6555 .....
6556 .....
6557 .....
6558 .....
6559 033214 BGNIST
6560 033214 SETPRI 0PRI05 ;ALLOW THE IYC TO INTERRUPT.
033214 012700 000240 ;MOV 0PRI05,R0
033220 104441 ;TRAP C1SPRI
6561 000024 INUM = INUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6562 033222 012767 000024 147074 MOV 0TNUM,TSTNUM ;SET UP THE TEST NUMBER. (20)
6563 033230 012767 177777 147050 MOV 0-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
6564 033236 012767 000001 150522 MOV 01,ERRTYP ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6565 033244 012767 003721 150516 MOV 02001,ERRMBR ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6566 033252 012767 012602 150512 MOV 0EM2001,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6567 .....
6568 ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6569 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6570 ; THIS SUBROUTINE REPORTS ERRORS ***** 2001 *****
6571 .....
6572 033260 004767 166072 JSR PC,RESET ;RESET THE DHU 11, REPORT ANY ERRORS FOUND.
6573 033264 103016 BCC 601 ;FATAL RESET ERROR? YES, EXIT THE TEST.
6574 .....
6575 ; READ THE STAT REGISTER ID BIT AND VERIFY THAT IT IS CLEAR.
6576 .....
6577 033266 017701 146756 MOV 0FSLSA,R1 ;READ THE STAT REGISTER CONTENTS.
6578 033272 032701 000400 BIT 0BIT8,R1 ;CHECK THE ID BIT.
6579 033276 001011 BNE 601 ;ID BIT SET? YES, EXIT THE TEST.
6580 033300 012767 003722 150462 MOV 02002,ERRMBR ;NO, SET THE ERROR REPORT NUMBER TO 2002.
6581 033306 012701 012652 MOV 0EM2002,R1 ;GET THE PROPER ERROR MESSAGE.
6582 033312 012767 016136 150454 MOV 0ER0503,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
6583 033320 ERROR ;ERROR NUMBER ***** 2002 *****
033320 104460 ;TRAP C1ERROR
6584 033322 005067 146760 601: CLR CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
6585 033326 ENDTST
033326 104401 ;L10051: TRAP C1ETST

```

```

6587 .SBTTL  HARDWARE TEST          TXENBI
6588 .....
6589 TX ENABLE (INACTIVE) TEST
6590 THIS TEST VERIFIES THAT WHEN THE LINE UNDER TEST'S TX ENABLE BIT IS
6591 CLEAR, TRANSMISSION WILL NOT TAKE PLACE ON THAT LINE.
6592 THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, AND ON ALL ACTIVE LINES.
6593 .....
6594
6595
6596 BGNTEST
6597 033330 012700 000240          SETPRI 0PRI05          ;ALLOW LTC INTERRUPTS.
6598 033330 104441          MOV 0PRI05,R0
6599 033330 000025          TRAP C1SPRI
6600 033336 012767 000025 146760  INUM = INUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6601 033344 012767 177777 146734  MOV 0TNUM,TSTNUM          ;SET UP THE TEST NUMBER. (23)
6602 033352 012767 000001 150406  MOV 01,C1RLCF          ;INDICATE THAT WE ARE IN A TEST.
6603 033360 012767 004375 150402  MOV 01,ERRTYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6604 033366 012767 012725 150376  MOV 02301,ERRNBR          ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
6605 033374 012767 017222 150372  MOV 0EM2301,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERRIBL.
6606                                MOV 0ER9101,ERRBLK          ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6607
6608 ; RESET THE OUT TO A KNOWN STATE. REMOVE THE STATUS CODES FROM THE FIFO.
6609 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6610 ; THIS SUBROUTINE REPORTS ERROR ***** 2301 *****
6611 033402 004767 164374          JSR PC,C1NRST          ;RESET THE DMU-11, REPORT ANY ERRORS FOUND.
6612                                BCC 601          ;RESET FAILURE?, ABORT THIS TEST.
6613
6614 ; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
6615 ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
6616 ; 2 STOP BITS.
6617 ; ENABLE TRANSMITTERS ON ALL LINES.
6618 033410 016705 146614          MOV ACTLNS,R5          ;PASS THE ACTIVE LINE BIT MAP.
6619 033414 012700 000200          MOV 0200,R0          ;PASS THE LNCtrl CONTENTS.
6620 033420 004767 170126          JSR PC,W1WUNC          ;INITIALISE THE LNCtrl REGISTERS.
6621 033424 012700 177670          MOV 017670,R0          ;PASS THE LPR CONTENTS.
6622 033430 004767 170146          JSR PC,W1W1PR          ;INITIALISE THE LPR REGISTERS ON ALL LINES.
6623 033434 012704 000012          MOV 010,R4          ;PASS DELAY TIME OF 10 MILLI-SECONDS.
6624 033440 004767 164460          JSR PC,DELAY          ;WAIT FOR LNCtr AND LPR REGS TO BE UPDATED.
6625 033444 012705 177777          MOV 0MAPLNS,R5          ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
6626 033450 004767 167224          JSR PC,TXENBI          ;ENABLE TRANSMITTERS ON ALL LINES.
6627
6628 ; TEST ALL ACTIVE LINES INDIVIDUALLY.
6629 ; DISABLE TRANSMISSION ON EACH ACTIVE LINE.
6630
6631 033454 012703 000001          MOV 01,R3          ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
6632 033460 005004          CLR R4          ;CLEAR THE LINE NUMBER COUNTER.
6633 033462 012767 004376 150300 23: MOV 02302,ERRNBR          ;SET THE ERROR NUMBER TO 2302.
6634 033470 010367 146534          BIT R3,ACTLNS          ;CHECK IF THE LINE IS ACTIVE.
6635 033474 001453          BEQ 61          ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
6636
6637 ; CLEAR THE TX ENABLE BIT IN THE FAD2 REGISTER.
6638 ; SELECT THE LINE UNDER TEST.
6639 ; VERIFY IT IS CLEAR, REPORT ERROR IF SET.
6640

```

```

6641 033476 010305      MOV    R3,R5      ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6642 033500 004767 167100 JSR    PC,IXDSBL  ;DISABLE TRANSMISSION ON THE LINE UNDER TEST.
6643 033504 010477 146532 MOV    R4,0CSRA  ;SELECT THE LINE CURRENTLY UNDER TEST.
6644 033510 005777 146542 IST    0TXAD2A    ;VERIFY THE TX_ENABLE BIT IS SET.
6645 033514 100433      BMI     4$      ;GO REPORT ERROR IF TX_ENABLE BIT SET.
6646
6647      ; WRITE DATA BYTE (ASCII <LF>) TO THE OUTPUT FIFO.
6648      ; WAIT FOR A TX ACTION TO BE RETURNED, REPORT ERROR IF A TX ACTION
6649      ; IS FOUND BEFORE TIME OUT OCCURS.
6650
6651 033516 012767 004377 150244 MOV    02303,,ERRNBR ;SET ERROR NUMBER TO 2303.
6652 033524 112777 000012 146516 MOVB   012,0FDATA    ;WRITE THE DATA BYTE TO THE DUT'S OUTPUT FIFO.
6653 033532 012701 170003      MOV    0170003,R1    ;TEST BIT 15, TIMEOUT OF 3 MILLI SECS.
6654 033536 016702 146500      MOV    CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6655 033542 004767 167522 JSR    PC,WAIBIS    ;WAIT FOR TX ACTION TO COME BACK.
6656 033546 103416      BCS     4$      ;GO REPORT ERROR IF A TX-ACTION FOUND.
6657
6658      ; WAIT FOR THE DATA TO APPEAR IN THE FIFO, REPORT ERROR IF DATA FOUND.
6659
6660 033550 005267 150214      INC     ERRNBR      ;SET ERROR NUMBER TO 2304.
6661 033554 012701 070012      MOV    070012,R1    ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
6662 033560 016702 146456      MOV    CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6663 033564 004767 167500 JSR    PC,WAIBIS    ;WAIT FOR RX DATA AVAILABLE TO SET.
6664 033570 103405      BCS     4$      ;REPORT ERROR IF DATA RECEIVED IN THE FIFO.
6665 033572 005267 150172      INC     ERRNBR      ;SET ERROR NUMBER TO 2305.
6666 033576 017702 146442      MOV    0RBUFA,R2    ;READ THE DATA FROM THE FIFO.
6667 033602 100010      BPL     6$      ;SKIP ERROR REPORT IF DATA ISN'T THERE.
6668
6669 033604 010401      4$: MOV    R4,R1      ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6670 033606 012702 012772      MOV    0EM2302,R2    ;PASS THE MESSAGE TO BE REPORTED.
6671
6672 033612      ERROR      ;TX_ENABLE BIT BAD ON LINE: NN.
6673 033612 104460      ; ***** ERROR *****
6674
6675 033614 032767 000100 146402      BIT     0B1T06,OPTION ;EXIT THE TEST, WITH THE TEST FAILED MESSAGE.
6676 033622 001406      BEQ     60$      ;IF EXTENDED ERROR REPORTING HAS NOT BEEN
6677
6678
6679      ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6680
6681 033624 000241      6$: CLC
6682 033626 006103      ROL     R3
6683 033630 005204      INC     R4
6684 033632 020427 000020      CMP    R4,0NMLNS    ;HAVE ALL THE LINES BEEN TESTED?
6685 033636 002711      BLT     7$      ;NO! BRANCH TO TEST THE NEXT LINE.
6686
6687 033640 005067 146442      60$: CLR     CTRLCF    ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6688 033644
        033644 104401

```

L10052: TRAP C\$TEST

```

6690 .SBLT1 HARDWARE TEST - TXENBA-
6691
6692
6693
6694
6695
6696
6697
6698
6699 033646 BGNTEST
033646
6700 033646 SETPRI 0PRI05 ;ALLOW LTC INTERRUPTS. T22::
033646 012700 000240 MOV 0PRI05,R0
033652 104441 TRAP C1SPRI
6701 000026 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6702 033654 012767 000026 146442 MOV 0TNUM,TSTNUM ;SET UP THE TEST NUMBER. (24)
6703 033662 012767 177777 146416 MOV 01,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
6704 033670 012767 000001 150070 MOV 01,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6705 033676 012767 004541 150064 MOV 02401,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
6706 033704 012767 013030 150060 MOV 0EM2401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTAB.
6707 033712 012767 017222 150054 MOV 0ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6708
6709 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
6710 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6711 ; THIS SUBROUTINE REPORTS ERROR >>>> 2401 <<<<.
6712
6713 033720 004767 164056 JSR PC,CLRST ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6714 033724 103133 BCC 604 ;RESET FAILURE?, ABORT THIS TEST.
6715
6716 ; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
6717 ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
6718 ; 2 STOP BITS.
6719 ; DISABLE TRANSMITTERS ON ALL LINES.
6720
6721 033726 016705 146276 MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
6722 033732 012700 000200 MOV 0200,R0 ;PASS THE LNCtrl CONTENTS.
6723 033736 004767 167610 JSR PC,WTWLN ;INITIALISE THE LNCtrl REGISTERS.
6724 033742 012700 177670 MOV 0177670,R0 ;PASS THE LPR CONTENTS.
6725 033746 004767 167670 JSR PC,WTWLP ;INITIALISE THE LPR REGISTERS ON ALL LINES.
6726 033752 012704 000012 MOV 010,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
6727 033756 004767 164142 JSR PC,DELAY ;WAIT FOR LNCtrl AND LPR REGS TO BE UPDATED.
6728 033762 012705 177777 MOV 0MAPLNS,R5 ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
6729 033766 004767 166612 JSR PC,TXDSBL ;DISABLE TRANSMITTERS ON ALL LINES.
6730
6731 ; TEST ALL ACTIVE LINES INDIVIDUALLY.
6732 ; ENABLE TRANSMISSION ON EACH ACTIVE LINE.
6733
6734 033772 012703 000001 MOV 01,R3 ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
6735 033776 005004 CLR R4 ;CLEAR THE LINE NUMBER COUNTER.
6736 034000 012767 004542 147762 24: MOV 02402,ERRNBR ;SET THE ERROR NUMBER TO 2402.
6737 034006 030367 146216 BIT R3,ACTLNS ;CHECK IF THE LINE IS ACTIVE.
6738 034012 001467 BEQ 81 ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
6739
6740 ; SELECT THE LINE UNDER TEST.
6741 ; SET THE TX_ENABLE BIT IN TBU#A02 REGISTER.
6742 ; VERIFY IT IS SET, REPORT ERROR IF CLEAR.
6743

```

```

6744 034014 010305      MOV      R3,R5      ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6745 034016 004767 166656 JSR      PC,TXENBL    ;ENABLE TRANSMISSION ON THE LINE UNDER TEST.
6746 034022 012705 000012 MOV      R5,R5      ;SET TXCHAR/LOOP COUNT TO 10.
6747 034026 010477 146210 MOV      R4,RCSRA    ;SELECT THE LINE CURRENTLY UNDER TEST.
6748 034032 005777 146220 IST      BTXAD2A      ;VERIFY THE TX_ENABLE BIT IS SET.
6749 034036 100045      BPL      6$      ;GO REPORT ERROR IF TX_ENABLE BIT CLEAR.
6750
6751      ; WRITE DATA BYTE (ASCII <LF>) TO OUTPUT FIFO.
6752      ; WAIT FOR A TX_ACTION TO BE RETURNED, REPORT ERROR IF NO TX_ACTION
6753      ; FOUND BEFORE TIME-OUT OCCURS.
6754
6755 034040 012767 004543 147722 4$: MOV      R2403,ERRNBR ;SET ERROR NUMBER TO 2403.
6756 034046 112777 000012 146174 MOVB     R12,RDATA    ;WRITE THE DATA BYTE TO THE DUT'S OUTPUT FIFO.
6757 034054 012701 170004      MOV      R170004,R1    ;TEST BIT 15, TIMEOUT OF 4 MILLI SECS.
6758 034060 016702 146156      MOV      CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6759 034064 004767 167200      JSR      PC,WAIBIS    ;WAIT FOR TX_ACTION TO COME BACK.
6760 034070 103030      BCC      6$      ;GO REPORT ERROR IF NO TX-ACTION FOUND.
6761
6762      ; WAIT FOR THE DATA TO APPEAR IN THE FIFO, REPORT ERROR IF TIME-OUT.
6763
6764 034072 005267 147672      INC      ERRNBR      ;SET ERROR NUMBER TO 2404.
6765 034076 012701 070012      MOV      R70012,R1    ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
6766 034102 016702 146134      MOV      CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6767 034106 004767 167156      JSR      PC,WAIBIS    ;WAIT FOR RX_DATA_AVAILABLE TO SET.
6768 034112 103017      BCC      6$      ;REPORT ERROR IF NO DATA RECEIVED IN THE FIFO.
6769 034114 005267 147650      INC      ERRNBR      ;SET ERROR NUMBER TO 2405.
6770 034120 017702 146120      MOV      RRAUFA,R2    ;READ THE DATA FROM THE FIFO.
6771 034124 100012      BPL      6$      ;GO REPORT ERROR IF THERE IS'NT ANY DATA THERE.
6772 034126 005267 147636      INC      ERRNBR      ;SET ERROR NUMBER TO 2406.
6773 034132 000302      SWAB     R2      ;PUT THE LINE NUMBER IN THE LOW BYTE.
6774 034134 042702 177760      BIC      R177760,R2    ;CLEAR THE UNWANTED BITS.
6775 034140 020204      CMP      R2,R4      ;DID THE DATA COME FROM THE CORRECT LINE?.
6776 034142 001003      BNE      6$      ;NO; GO REPORT THE ERROR.
6777 034144 005305      DEC      R5      ;DECREMENT THE TXCHAR/LOOP COUNTER.
6778 034146 001334      BNE      4$      ;LOOP TO TX THE NEXT CHAR.
6779 034150 000410      BR      8$      ;GO TEST THE NEXT LINE.
6780
6781 034152 010401      MOV      R4,R1      ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6782 034154 012702 012772      MOV      R2EM2302,R2    ;PASS THE MESSAGE TO BE REPORTED.
6783
6784 034160      ERROR      ; "TX_ENABLE BIT BAD ON LINE: NN".
6785 034160 104460      ; >>>> ERROR <<<<.
6786      TRAP      C$ERROR
6787
6788
6789
6790      ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6791
6792 034172 010305      MOV      R3,R5      ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6793 034174 004767 166404 JSR      PC,TXD5BL    ;CLEAR THE TX_ENABLE BIT ON THIS LINE.
6794 034200 000241      CLC      ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
6795 034202 006103      ROL      R5      ;SHIFT THE BIT MAP FOR THE NEXT LINE.
6796 034204 005204      INC      R4      ;INCREMENT THE LINE NUMBER COUNTER.
6797 034206 020427 000020      CMP      R4,RNUMLNS ;HAVE ALL THE LINES BEEN TESTED?.
6798 034212 002672      BLT      2$      ;NO; BRANCH TO TEST THE NEXT LINE.
6799

```


614

C:\DHUBO\DHU-11\FUNC TEST PART1 MACRO M1200 15-MAR-84 09:15 PAGE 121-2
HARDWARE TEST - IXENBA

SEQ 175

6800 034214 005067 146066
6801 034220
034220
034220 104401

60\$: CLR CIRC
ENDTST

INDICATE THAT WE ARE NOT WITHIN A TEST.

1.10053: TRAP C\$ETST

```

6803          .SBTTL  HARDWARE TEST          - INTA -
6804          ;** *****
6805          ;*          - INTERRUPT TEST -
6806          ;*          THIS TEST VERIFIES THAT THE DEVICE UNDER TEST (DUT) WILL GENERATE
6807          ;*          RECEPTION AND TRANSMISSION INTERRUPTS CORRECTLY.  THIS TEST DOES
6808          ;*          NOT DEPEND ON THE USE OF THE SERIAL LINE TRANSMISSION OR RECEPTION
6809          ;*          CAPABILITIES OF THE DUT.  THE LINES ARE PUT IN INTERNAL LOOPBACK
6810          ;*          TO MINIMIZE ANY EXTERNAL EFFECTS THAT COULD BE CAUSED ON DEVICES
6811          ;*          ATTACHED TO THE SERIAL LINES.
6812          ;*          *****
6813          ;-- *****
6814
6815 034222      BGNTEST
6816 034222      SETPRI  @PRI05          ;ALLOW THE LTC TO INTERRUPT.
6817 034222      012700 000240          ;
6818 034222      034226 104441          ;
6819 034222      000027          ;
6820 034222      TNUM ** TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6821 034222      MOV  @TNUM,TSTNUM      ;SET UP THE TEST NUMBER. (26)
6822 034222      MOV  @-1,CTRLCF        ;INDICATE THAT WE ARE IN A TEST.
6823 034222      MOV  @1,ERRTYP          ;SET ERROR FATAL ERROR TYPE IN ERROR TABLE.
6824 034222      MOV  @!601,,ERRNBR      ;SET FIRST ERROR REPORT NUMBER IN ERROR TABLE.
6825 034222      MOV  @EM2601,,ERRMSG    ;SET TEST ERROR MESSAGE IN ERROR TABLE.
6826
6827          ;
6828          ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6829          ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6830          ; THIS SUBROUTINE REPORTS ERRORS FROM >>>> 2601 THRU 2602 <<<<.
6831
6832          ;--
6833          JSR  PC,RESETT              ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6834          BCS  2$                    ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
6835          JMP  60$                    ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
6836          MOV  @2603,,ERRNBR        ;SET THE ERROR REPORT NUMBER TO 2603.
6837
6838          ;
6839          ; ENABLE TRANSMITTERS ON ALL LINES.
6840
6841          ;--
6842          4$:  MOV  @MAPLNS,R5          ;PASS ACTIVE LINE BIT MAP.
6843          JSR  PC,TXENBL              ;ENABLE TRANSMISSION ON ALL LINES.
6844
6845          ;
6846          ; TEST RECEPTION INTERRUPTS.
6847          ; SET UP FOR RX AND TX INTERRUPTS:
6848          ; RX INTERRUPT SERVICE ROUTINE INPUTS A CHAR AND COUNTS THE INTERRUPT.
6849          ; TX INTERRUPT SERVICE ROUTINE COUNTS TX INTERRUPTS.
6850
6851          ;--
6852          CLR  RXINTC                  ;CLEAR THE RX INTERRUPT COUNTER.
6853          CLR  RXINTF                  ;CLEAR THE RX INTERRUPT FLAGS.
6854          CLR  TXINTC                  ;CLEAR THE TX INTERRUPT COUNTER.
6855          MOV  @BUF BAS,BUF PTR        ;LOAD THE BUFFER PTR WITH THE BUFFER BASE ADR.
6856          SETVEC RXVECA,@RXINPT,@PRI06 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
6857
6858          MOV  @PRI06,(SP)
6859          MOV  @RXINPT,(SP)
6860          MOV  RXVECA,(SP)
6861          MOV  @3,(SP)
6862          TRAP C$SVEC
6863          ADD  @10,SP
6864
6865          SETVEC TXVECA,@CACHIX,@PRI06 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
6866
6867          MOV  @PRI06,(SP)
6868          MOV  @CACHIX,(SP)

```


034376	016746	145632				MOV	TXVECA, -(SP)
034402	012746	000003				MOV	#3, -(SP)
034406	104437					TRAP	C\$SVEC
034410	062706	000010				ADD	#10, SP
6849 034414					SETPRI	#PRI04	;ALLOW DEVICE INTERRUPTS.
034414	012700	000200				MOV	#PRI04, R0
034420	104441					TRAP	C\$SPRI
6850					;;		
6851					;;		ENABLE RECEPTION INTERRUPTS.
6852					;;		DELAY 4 MS TO ALLOW TIME FOR THE INTERRUPTS TO TAKE PLACE.
6853					;;		DISABLE RECEPTION INTERRUPTS.
6854					;;		
6855 034422	004767	165624			JSR	PC, RXIE1	;ENABLE THE RECEPTION INTERRUPTS.
6856 034426	012704	000004			MOV	#4, R4	;PASS 4 MS COUNT TO THE DELAY ROUTINE.
6857 034432	004767	163466			JSR	PC, DELAY	;DELAY 4 MILLI-SECONDS.
6858 034436	004767	165550			JSR	PC, RXIE0	;DISABLE RECEPTION INTERRUPTS.
6859					;;		
6860					;;		VERIFY THAT THE CORRECT INTERRUPTS TOOK PLACE.
6861					;;		TEST THE INT COUNTER TO VERIFY THAT INTERRUPTS TOOK PLACE.
6862					;;		
6863 034442	005767	145646			TST	RXINTC	;CHECK THE RX INTERRUPT COUNT.
6864 034446	001017				BNE	#	;SKIP THE FOLLOWING ERRORS IF COUNT <> 0.
6865					;;		
6866					;;		DETERMINE REASON FOR NO RX INTERRUPTS AND PRINT PROPER ERROR MESSAGE.
6867					;;		
6868 034450	012701	013310			MOV	#EM2604, R1	;SET UP MSG IN CASE "RX, DATA, AVAIL IS CLR".
6869 034454	032777	000200	145560		BIT	#BIT7, #CSRA	;TEST THE RX, DATA, AVAIL BIT OF THE CSR.
6870 034462	001416				BEQ	#	;GO REPORT ERROR IF RX, DATA, AVAIL IS CLR.
6871 034464	012701	013222			MOV	#EM2603, R1	;SET UP MSG IN CASE "DATA, VALID IS CLEAR".
6872 034470	032777	100000	145546		BIT	#BIT15, #RBUFA	;TEST THE DATA, VALID BIT OF THE FIFO.
6873 034476	001410				BEQ	#	;GO REPORT ERROR IF DATA, VALID IS CLEAR.
6874 034500	012701	013131			MOV	#EM2602, R1	;SET UP MSG, "DATA, VALID IS SET".
6875 034504	000405				BR	#	;GO REPORT THE ERROR.
6876					;;		
6877					;;		IF RX INTS OCCURRED WITH RX, DATA, AVAIL CLEAR, REPORT THE ERROR.
6878					;;		
6879 034506	005767	145604			6\$:	TST	RXINTF
6880 034512	100014					BPL	#10
6881 034514	012701	013404				MOV	#EM2605, R1
6882							;CHECK THE RX INTERRUPT FLAGS.
6883							;SKIP THE ERROR IF FLAG 1 IS CLEAR.
6884							;SET UP THE PROPER MESSAGE.
6885 034520					8\$:		
034520	104455					ERRDF	2603, EM2601, ER0503;
034522	005053						;>>>> ERROR #2603 <<<<.
034524	013073						TRAP
034526	016136						3\$ERDF
6886							.WORD
6887							2603
6888							.WORD
6889							EM2601
6890							.WORD
6891 034530	052767	000100	145466				ER0503
6892 034536	001002						
6893 034540	000167	000556					
6894							
6895							

;;
 ;;EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
 ;;
 BIT #BIT06, OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
 BNE # ;
 JMP 34\$;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
 ;;

```

6896      ; VERIFY THAT NO TX INTERRUPTS HAVE BEEN GENERATED SO FAR IN THIS TEST.
6897      ;
6898 034544 016702 145556      10$: MOV TXINTC,R2      ;LOAD # OF TX INTERRUPTS FOR ER0504 RTN.
6899 034550 001414      BEQ 12$      ;SKIP ERROR IF NO TX INTERRUPTS.
6900      ;REPORT "TX INTERRUPTS(S) RECEIVED WITH TX INTERRUPTS DISABLED."
6901 034552 012701 007505      MOV #EM0526,R1      ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
6902 034556      ERDF 2604,EM2601,ER0504;      >>>> ERROR #2604 <<<<.
        034556 104455      TRAP C$ERDF
        034560 005054      .WORD 2604
        034562 013073      .WORD EM2601
        034564 016174      .WORD ER0504
6903
6904      ;
6905      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
6906      ;
6907      ;
6908 034566 032767 000100 145430      BIT #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
6909 034574 001002      BNE .+6      ;
6910 034576 000167 000520      JMP 34$      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
6911
6912      ;
6913      ; CLEAN OUT THE INTERRUPT VECTORS USED IN THIS TEST.
6914      ;
6915 034602      12$: SETPRI #PRI06      ;DISABLE DEVICE INTERRUPTS.
        034602 012700 000300      MOV #PRI06,R0
        034606 104441      TRAP C$SPRI
6916 034610      CLRVEC RXVECA      ;RETURN RX INT VECTOR TO UNUSED POOL
        034610 016700 145416      MOV RXVECA,R0
        034614 104436      TRAP C$CVEC
6917 034616      CLRVEC TXVECA      ;RETURN TX INT VECTOR TO UNUSED POOL
        034616 016700 145412      MOV TXVECA,R0
        034622 104436      TRAP C$CVEC
6918
6919      ;
6920      ; TEST TRANSMISSION INTERRUPTS.
6921      ; SET UP FOR RX AND TX INTERRUPTS:
6922      ; RX INTERRUPT SERVICE ROUTINE COUNTS RX INTERRUPTS.
6923      ; TX INTERRUPT SERVICE ROUTINE COUNTS THE INTERRUPT AND SETS FLAGS.
6924 034624 005067 145464      CLR RXINTC      ;CLEAR THE RX INTERRUPT COUNTER.
6925 034630 005067 145472      CLR TXINTC      ;CLEAR THE TX INTERRUPT COUNTER.
6926 034634 005067 145470      CLR TXINTF      ;CLEAR THE RX INTERRUPT FLAGS.
6927 034640      SETVEC RXVECA,#CACHRX,#PRI06 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
        034640 012746 000300      MOV #PRI06,(SP)
        034644 012746 023632      MOV #CACHRX,(SP)
        034650 016746 145356      MOV RXVECA,-(SP)
        034654 012746 000003      MOV #3,(SP)
        034660 104437      TRAP C$SVEC
        034662 062706 000010      ADD #10,SP
6928 034666      SETVEC TXVECA,#TXINTR,#PRI06 ;SET UP INT VECTOR TO TX INT ROUTINE.
        034666 012746 000300      MOV #PRI06,(SP)
        034672 012746 024146      MOV #TXINTR,(SP)
        034676 016746 145332      MOV TXVECA,(SP)
        034702 012746 000003      MOV #3,(SP)
        034706 104437      TRAP C$SVEC
        034710 062706 000010      ADD #10,SP
6929 034714      SETPRI #PRI04      ;ALLOW DEVICE INTERRUPTS.
        034714 012700 000200      MOV #PRI04,R0

```

```

034720 104441                                TRAP    C$SPRI
6930
6931
6932
6933 034722 012705 000022
6934 034726 012701 000144
6935 034732 012702 100000
6936 034736 016704 145300
6937 034742 012703 100000
6938 034746 004767 163326
6939 034752 103026
6940 034754 005003
6941 034756 004767 163316
6942 034762 103005
6943 034764 005305
6944 034766 001365
6945
6946 034770 012701 013526
6947 034774 000402
6948 034776 012701 013622
6949 035002
      035002 104455
      035004 005055
      035006 013467
      035010 016136
6950
6951
6952
6953
6954 035012 032767 000100 145204
6955 035020 001540
6956 035022 004767 166006
6957 035026 000430
6958
6959
6960
6961 035030 004767 166000
6962 035034 012704 000062
6963 035040 004767 163060
6964 035044 005767 145256
6965 035050 001417
6966 035052 012701 013526
6967 035056 005767 145246
6968 035062 100002
6969 035064 012701 013673
6970
6971 035070
      035070 104455
      035072 005056
      035074 013467
      035076 016136
6972
6973
6974
6975
6976 035100 032767 000100 145116
6977 035106 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

```

```

6978
6979
6980
6981
6982 035110 005067 145212
6983 035114 005067 145210
6984
6985
6986
6987 035120 012705 177777
6988 035124 012700 000200
6989 035130 004767 166416
6990 035134 012700 156430
6991 035140 004767 166436
6992
6993
6994
6995 035144 016701 145140
6996 035150 005002
6997 035152 010177 145064
6998
6999 035156 112777 000000 145064
7000 035164 005201
7001 035166 005202
7002 035170 020227 000020
7003 035174 002766
7004
7005
7006
7007 035176 012704 000372
7008 035202 004767 162716
7009
7010
7011
7012 035206 005767 145114
7013 035212 001010
7014
7015
7016
7017 035214 012701 013752
7018 035220 005777 145016
7019 035224 100410
7020 035226 012701 014044
7021 035232 000405
7022
7023
7024
7025 035234 005767 145070
7026 035240 100012
7027 035242 012701 013670
7028
7029
7030
7031 035246
      035246 104455
      035250 005057
      035252 013467

;+
; PREPARE TX INTERRUPT COUNTER AND FLAGS.
;-
24$: CLR TXINTC ;CLEAR THE TX INTERRUPT COUNT.
      CLR TXINTF ;CLEAR THE TX INTERRUPT FLAGS.
;+
; SET UP LINE PARAMETERS FOR TRANSMISSION.
;-
      MOV 0MAPLNS,R5 ;PASS ACTIVE LINES BIT MAP.
      MOV 0200,R0 ;PASS INERT STATE, INTERNAL LOOPBACK.
      JSR PC,WTWLC ;DISABLE RECEPTION AND DMA, ETC. ON OUT.
      MOV 0156430,R0 ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
      JSR PC,WTWLP ;WRITE TO ALL LPR REGISTERS.
;+
; SEND A NULL CHAR TO EACH LINE.
;-
      MOV IESTAT,R1 ;SET UP THE STATE OF THE INTERRUPT ENABLE BITS.
      CLR R2 ;CLEAR THE LINE COUNTER.
25$: MOV R1,0CSRA ;SET UP THE LINE NUMBER AND INTERRUPT ENABLE
      ;BITS IN THE CSR.
      MOV 00,0FDATA ;SEND A NULL CHARACTER TO THE OUTPUT FIFO.
      INC R1 ;NEXT CSR CONTENTS.
      INC R2 ;NEXT LINE.
      CMP R2,0NUMLNS ;IF ALL LINES HAVE NOT BEEN SERVICED THEN
      BLT 25$ ;BRANCH.
;+
; DELAY 250 MILLI-SECONDS TO ALLOW INTERRUPTS TO OCCUR.
;-
      MOV 0250,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 0EM2610,R1 ;SET UP MSG IN CASE "TX_ACTION IS SET".
      TST 0CSRA ;CHECK THE OUT CSR.
      BMI 28$ ;GO TO REPORT ERROR IF TX_ACTION IS SET.
      MOV 0EM2611,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 0EM2609,R1 ;SET UP TX INT WITH "TX_ACTION CLR" MSG.
;+
; REPORT "TRANSMIT INTERRUPT TEST ERROR:...."
;-
28$: ERRDF 2607,EM2606,ER0503; >>>>> ERROR 02607 <<<<<,
      TRAP C$ERRDF
      .WORD 2607
      .WORD EM2606

```

```

035254 016136 .WORD ERO503
7032
7033
7034 ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7035 ;
7036 035256 032767 000100 144740 BIT 0BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
7037 035264 001411 BEQ 32$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7038
7039 ;
7040 ; VERIFY THAT NO RX INTERRUPTS HAVE BEEN GENERATED SO FAR IN THIS TEST.
7041 ;
7042 035266 016702 145022 30$: MOV RXINTC,R2 ;LOAD # OF RX INTERRUPTS FOR ERO504 RTN.
7043 035272 001406 BEQ 32$ ;SKIP ERROR IF NO RX INTERRUPTS.
7044 035274 012701 007415 MOV 0EM0525,R1 ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
7045 ;REPORT "RX INTERRUPTS(S) RECEIVED WITH RX INTERRUPTS DISABLED."
7046 035300 ERRDF 2608,EM2606,ERO504; >>>> ERROR #2608 <<<<.
035300 104455 TRAP C$ERDF
035302 005060 .WORD 2608
035304 013467 .WORD EM2606
035306 016174 .WORD ERO504
7047
7048 ; DISABLE INTERRUPTS AND CLEAN OUT THE INTERRUPT VECTORS USED IN THIS TEST.
7049 ;
7050 035310 005001 32$: CLR R1 ;CLEAR BOTH TRANSMITTER
7051 035312 004767 165456 JSR PC,TXIE0 ; INTERRUPT ENABLE AND RECEIVER
7052 035316 004767 164670 JSR PC,RXIE0 ; INTERRUPT ENABLE BITS IN THE OUT CSR.
7053 035322 012700 000300 34$: SETPRI 0PRI06 ;DISABLE DEVICE INTERRUPTS.
035326 104441 MOV 0PRI06,RO
TRAP C$SPRI
7054 035330 CLRVEC RXVECA ;RETURN RX INT VECTOR TO UNUSED POOL.
035330 016700 144676 MOV RXVECA,RO
035334 104436 TRAP C$CVEC
7055 035336 CLRVEC TXVECA ;RETURN TX INT VECTOR TO UNUSED POOL.
035336 016700 144672 MOV TXVECA,RO
035342 104436 TRAP C$CVEC
7056
7057 035344 005067 144736 60$: CLR CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7058 035350 SETPRI 0PRI07 ;DISABLE ALL INTERRUPTS.
035350 012700 000340 MOV 0PRI07,RO
035354 104441 TRAP C$SPRI
7059 035356 ENDTST
035356 104401 L10054: TRAP C$ETST
7060

```

```

7062 .SBTTL  HARDWARE TEST          BRLEVA -
7063 ;* *****
7064 ;*          BR LEVEL TEST 8 -
7065 ;*      THIS TEST VERIFIES THAT THE DEVICE UNDER TEST (DUT) WILL GENERATE
7066 ;*      RECEPTION AND TRANSMISSION INTERRUPTS AT THE CORRECT BR LEVEL.
7067 ;*      THIS TEST DOES NOT DEPEND ON THE USE OF THE SERIAL LINE TRANSMISSION
7068 ;*      OR RECEPTION CAPABILITIES OF THE DUT.  THE LINES ARE PUT IN INTERNAL
7069 ;*      LOOPBACK TO MINIMIZE ANY EXTERNAL EFFECTS THAT COULD BE CAUSED ON
7070 ;*      DEVICES ATTACHED TO THE SERIAL LINES.
7071 ;*
7072 ;* *****
7073 ;*
7074 035360          BGNTST
7075 035360          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.          T24::
7076 035360 012700 000240          MOV      #PRI05,R0
7077 035360 104441          TRAP      C$SPRI
7078 035366 000030          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
7079 035366 012767 000030 144730  MOV      #TNUM,T$NUM          ;SET UP THE TEST NUMBER. (30)
7080 035374 012767 177777 144704  MOV      #1,CTRLCF          ;INDICATE THAT WE ARE IN A TEST.
7081 035402 012767 000001 146356  MOV      #1,ERRTYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
7082 035410 012767 005671 146352  MOV      #3001,,ERRNBR          ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
7083 035416 012767 014223 146346  MOV      #EM3001,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
7084 035424 005067 145032          CLR      ERSRNF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
7085 ;*
7086 ;* RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
7087 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
7088 ;* THIS SUBROUTINE REPORTS ERRORS FROM >>>> 3001 THRU 3002 <<<<.
7089 ;*
7090 035430 004767 163726          JSR      PC,RESETT          ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
7091 035434 103402          RCS      2$          ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
7092 035436 000167 000644          JMP      60$          ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
7093 035442 012767 005673 146320 2$: MOV      #3003,,ERRNBR          ;SET THE ERROR REPORT NUMBER TO 3003.
7094 ;*
7095 ;* ENABLE TRANSMITTERS ON ALL LINES.
7096 035450 012705 177777          4$: MOV      #MAPLNS,R5          ;PASS ACTIVE LINE BIT MAP.
7097 035454 004767 165220          JSR      PC,TXENBL          ;ENABLE TRANSMISSION ON ALL LINES.
7098 ;*
7099 ;* GENERATE A TRANSMISSION INTERRUPT REQUEST.
7100 ;* PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING INTS.
7101 035460          SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
7102 035460 012700 000340          MOV      #PRI07,R0
7103 035464 104441          TRAP      C$SPRI
7104 035466          SETVEC  TXVECA,#TXINTR,#PRI07 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
7105 035466 012746 000340          MOV      #PRI07,(SP)
7106 035472 012746 024146          MOV      #TXINTR,(SP)
7107 035476 016746 144532          MOV      TXVECA,(SP)
7108 035502 012746 000003          MOV      #3,(SP)
7109 035506 104437          TRAP      C$SVEC
7110 035510 062706 000010          ADD      #10,SP
7111 ;*
7112 ;* SET UP DUT FOR TRANSMISSION INTERRUPTS;
7113 ;* SET UP INTERNAL LOOPBACK.
7114 ;* SET UP LINE PARAMETERS FOR TRANSMISSION.
7115 ;*

```


B15

C70HUB0 DHU 11 FUNC 1ST PART1 MACRO M1200 15-MAR 84 09:15 PAGE 123-1
HARDWARE TEST - BRLEVA -

SEQ 187

```

7108 035514 012705 177777      MOV    @MAPLNS,R5      ;PASS ACTIVE LINES BIT MASK.
7109 035520 012700 000200      MOV    #200,R0        ;PASS INERT STATE, INTERNAL LOOPBACK.
7110 035524 004767 166022      JSR     PC,WTWUNC      ;DISABLE RECEPTION AND DMA, ETC. ON DUT.
7111 035530 012700 156430      MOV    #156430,R0      ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
7112 035534 004767 166042      JSR     PC,WTWLPR      ;WRITE INTO ALL LPR REGISTERS.
7113
7114      ; SEND A NULL CHAR TO EACH LINE.
7115
7116 035540 016701 144544      MOV    IESTAT,R1      ;SET UP THE STATE OF THE INTERRUPT ENABLE BITS.
7117 035544 010177 144472      51:    MOV    R1,DCSRA      ;SET UP THE LINE NUMBER AND INTERRUPT ENABLE
                                ;BITS IN THE CSR.
7118
7119 035550 112777 000000 144472      MOVB   #0,DFDATA      ;SEND A NULL CHARACTER TO THE OUTPUT FIFO.
7120 035556 005201 000000      INC     R1              ;NEXT LINE.
7121 035560 020127 000020      CMP    R1,#NUMLNS     ;IF ALL LINES HAVE NOT BEEN SERVICED THEN
7122 035564 002767 000000      BLT     51              ;BRANCH.
7123
7124      ; DELAY 50 MS TO ALLOW TIME FOR THE INTERRUPT TO BE GENERATED.
7125
7126 035566 012704 000062      MOV    #50,R4        ;PASS 50 MS TIME TO THE DELAY ROUTINE.
7127 035572 004767 162326      JSR     PC,DELAY      ;DELAY 50 MILLI-SECONDS.
7128
7129      ; GENERATE A RECEPTION INTERRUPT REQUEST.
7130
7131 035576      SETVEC  RXVECA,@RXBRR1,@PRIO7 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
                                MOV    @PRIO7,(SP)
                                MOV    @RXBRR1,(SP)
                                MOV    RXVECA,(SP)
                                MOV    #3,(SP)
                                TRAP    C$VEC
                                ADD     #10,SP
7132 035576 012746 000340
7133 035602 012746 023756
7134 035606 016746 144420
7135 035612 012746 000003
7136 035616 104437
7137 035620 062706 000010
7138
7139      ; SET UP FOR THE LOOP WHICH TESTS THE INTERRUPT BR LEVELS.
7140
7141
7142 035624 012705 000340      MOV    #540,R5      ;SET UP THE PRIORITY LEVEL TO 7.
7143 035630 005003      CLR    R5          ;CLEAR THE RX PRIORITY STORE AND FLAGS.
7144 035632 005002      CLR    R2          ;CLEAR THE TX PRIORITY STORE AND FLAGS.
7145
7146      ; ENABLE TX AND RX INTERRUPTS.
7147      ; PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING THE INTERRUPTS.
7148
7149 035634 004767 164412      JSR     PC,RXIE1      ;ENABLE RECEIVER INTERRUPTS.
7150 035640 004767 165170      JSR     PC,TXIE1      ;ENABLE TRANSMITTER INTERRUPTS.
7151
7152      ; LOOP, LOWERING THE PROCESSOR PRIORITY UNTIL THE OUT INTERRUPTS ON RX AND TX.
7153
7154      61:    CLR    TXINTC      ;CLEAR THE TX INTERRUPT COUNTER.
7155      CLR    TXINTF      ;CLEAR THE TX INTERRUPT FLAGS.
7156      CLR    RXINTC      ;CLEAR THE RX INTERRUPT COUNTER.
7157      CLR    RXINTF      ;CLEAR THE RX INTERRUPT FLAGS.
7158      GETPRI  R5          ;SET PROCESSOR PRIORITY TO THE SELECTED VALUE.
                                MOV    R5,R0
                                TRAP    C$PRI
7159
7160      MOV    #1,R4        ;PASS 1 MS COUNT TO THE DELAY ROUTINE.
7161      JSR     PC,DELAY      ;DELAY 1 MS TO ALLOW INTERRUPTS TO OCCUR.
7162
7163      ; DETERMINE IF ANY RX OUT INTERRUPTS OCCURRED.
7164      ; LOG THE PROCESSOR PRIORITY FOR THE RX INTERRUPT IF FIRST RX INT.

```

C15

CZDMU10 DHU 11 FUNC 1ST PART1 MACRO M1200 15-MAR 84 09:15 PAGE 123 2
HARDWARE TEST - BRIEVA

GEQ 184

```

7157
7158 035700 005767 144410
7159 035704 001412
7160
7161
7162
7163 035706 005703
7164 035710 001010
7165 035712 010503
7166 035714 052703 100000
7167 035720 016700 144372
7168 035724 042700 137777
7169 035730 050003
7170
7171
7172
7173
7174 035732 005767 144370
7175 035736 001405
7176
7177
7178
7179 035740 005702
7180 035742 100403
7181 035744 010502
7182 035746 052702 100000
7183
7184
7185
7186
7187 035752 162705 000040
7188 035756 002402
7189 035760 030203
7190 035762 100330
7191
7192
7193
7194 035764
035764 012700 000340
035770 104441
7195 035772
035772 016700 144234
035776 104436
7196 036000
036000 016700 144230
036004 104436
7197
7198
7199
7200
7201
7202
7203 036006 005702
7204 036010 100420
7205
7206
7207

;
; CHECK THE RECEIVE INTERRUPT COUNTER.
; SKIP THE PRIORITY LOG IF NO RX INT OCCURRED.
;
; IF THIS IS THE FIRST RX INTERRUPT, LOG THE PRIORITY.
;
; CHECK THE RX PRIORITY STORE AND FLAGS.
; GOTO TEST FOR TX INTS IF NOT THE FIRST RX INT.
; LOG THE PRESENT PRIORITY IN THE RX PRIO STORE.
; SET THE RX INT HAS OCCURRED FLAG.
; GET THE RX INTERRUPT ROUTINE FLAGS.
; CLEAR ALL BUT THE TX INT ERROR FLAG.
; IF TX INT ERROR, SET BIT 14 OF THE PRIO FLAGS.
;
; DETERMINE IF ANY TX OUT INTERRUPTS HAVE OCCURRED.
; LOG THE PRESENT PROCESSOR PRIORITY IF THIS IS THE FIRST TX INTERRUPT.
;
; CHECK THE TRANSMIT INTERRUPT COUNTER.
; SKIP THE PRIORITY LOG IF NO TX INT OCCURRED.
;
; IF THIS IS THE FIRST TX INTERRUPT, LOG THE PRIORITY.
;
; CHECK THE TX PRIORITY STORE AND FLAGS.
; SKIP THE LOGGING IF NOT FIRST TX INTERRUPT.
; LOG THE PRESENT PRIORITY IN THE TX PRIO STORE.
; SET THE TX INT HAS OCCURRED FLAG.
;
; SELECT NEXT PROCESSOR PRIORITY.
; TEST FOR BOTH RX AND TX INTERRUPTS HAVING OCCURRED, LOOP IF NOT.
;
; DECREMENT PRIORITY LEVEL BY ONE.
; GOTO CHECK FOR ERRORS IF BELOW PRIORITY ZERO.
; AND PRIO FLAGS TOGETHER, ALTER NONE OF THEM.
; LOOP IF RX AND TX INTS HAVEN'T BOTH OCCURRED.
;
; DISABLE INTERRUPTS AND CLEAR INTERRUPT VECTORS.
;
; DISABLE ALL INTERRUPTS.
;
; RETURN RX INT VECTOR TO UNUSED POOL.
;
; RETURN TX INT VECTOR TO UNUSED POOL.
;
; DETERMINE WHETHER TX INT OCCURRED OR NOT.
; SKIP THESE ERRORS IF TX INT OCCURRED.
;
; DETERMINE REASON THAT NO TX INT OCCURRED.
;

```



```

7208 036012 012701 013752      MOV    0EM2610,R1      ;SELECT "NO TX INT FROM TX ACTION" MESSAGE.
7209 036016 005777 144220      LST     0CSR          ;CHECK THE TX ACTION BIT OF THE DUT CSR.
7210 036022 100402              BMI     14$              ;SKIP TX ACTION CLR MSG SELECTION IF IT IS SET.
7211 036024 012701 014044      MOV     0EM2611,R1      ;SELECT "TX ACTION CLEAR AFTER CHARS SENT" MSG.
7212              ;REPORT "INTERRUPT BR LEVEL TEST ERROR:"
14$:  ERRDF  3003,EM3001,ER0503;      >>>> ERROR 03003 <<<<.
                                TRAP    C$ERRDF
                                .WORD   3003
                                .WORD   EM3001
                                .WORD   ER0503
7213 036030              ;
036030 104455              ;
036032 005673              ;
036034 014223              ;
036036 016136              ;
7214
7215
7216
7217
7218 036040 032767 000100 144156  ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7219 036046 001513              ;
                                BIT     0BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
                                BEQ     26$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7220
7221 036050 000427              ;
                                BR      18$              ;SKIP THE BR LEVEL CHECK. NO TX INT OCCURRED.
7222
7223              ; VERIFY THAT THE TX INTERRUPT WAS AT THE PROPER BR LEVEL.
7224
16$:  MOV     R2,R4              ;CALCULATE THE BR LEVEL
                                BIC     0177400,R4      ; THAT THE TRANSMIT
                                ASR     R4              ; INTERRUPT WAS
                                ASR     R4              ; REQUESTED AT, WHICH
                                ASR     R4              ; IS ONE GREATER THAN
                                ASR     R4              ; THE PROCESSOR PRIORITY.
                                ASR     R4              ; LEVEL AT WHICH THE
                                INC     R4              ; TRANSMIT INTERRUPT OCCURRED.
                                MOVB    BRLEV1,R5        ;GET THE EXPECTED INTERRUPT BR LEVEL.
                                CMPB    R4,R5          ;COMPARE THE INTERRUPT BR LEVEL WITH EXPECTED.
                                BEQ     18$              ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
                                ;REPORT "TX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
7225 036052 010204              MOV     0EM3003,R1      ;SELECT THE ERROR MESSAGE FOR THE ERROR CALL.
7226 036054 042704 177400      ERRDF  3004,EM3001,ER3001;      >>>> ERROR 03004 <<<<.
7227 036060 006204              ;
7228 036062 006204              ;
7229 036064 006204              ;
7230 036066 006204              ;
7231 036070 006204              ;
7232 036072 005204              ;
7233 036074 116705 144140      TRAP    C$ERRDF
7234 036100 120405              .WORD   3004
7235 036102 001412              .WORD   EM3001
7236              .WORD   ER3001
7237 036104 012701 014355      ;
7238 036110              ;
036110 104455              ;
036112 005674              ;
036114 014223              ;
036116 016620              ;
7239
7240
7241
7242 036120 032767 000100 144076  ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7243 036126 001463              ;
                                BIT     0BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
                                BEQ     26$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7244
7245
7246
7247              ; DETERMINE IF RX INTERRUPT OCCURRED.
18$:  LST     R3              ;CHECK THE RX INT OCCURRED FLAG.
                                BMI     22$              ;SKIP THESE ERRORS IF RX INT OCCURRED.
7248 036130 005703              ;
7249 036132 100421              ;
7250
7251              ; DETERMINE REASON THAT NO RX INT OCCURRED.
7252
7253 036134 012701 014127      MOV     0EM2612,R1      ;SELECT "NO RX INT FROM RX DATA AVAIL" MSG.
7254 036140 032777 000200 144074  ;CHECK THE RX DATA AVAIL BIT OF THE DUT CSR.
7255 036142 001002              BMI     20$              ;SKIP RX DATA AVAIL CLR MSG IF BIT IS SET.
7256 036150 012701 014262      MOV     0EM3002,R1      ;SELECT "NO RX DATA AVAIL AFTER RESET" MSG.

```

```

7257
7258 036154 104455
      036156 005675
      036160 014223
      036162 016136

7259
7260
7261
7262 036164 032767 000100 144032
7263 036172 001441
7264
7265 036174 000427
7266
7267
7268
7269 036176 010304
7270 036200 042704 177400
7271 036204 006204
7272 036206 006204
7273 036210 006204
7274 036212 006204
7275 036214 006204
7276 036216 005204
7277 036220 116705 144014
7278 036224 120405
7279 036226 001412
7280
7281 036230 012701 014432
7282 036234
      036234 104455
      036236 005676
      036240 014223
      036242 016620

7283
7284
7285
7286 036244 032767 000100 143752
7287 036252 001411
7288
7289
7290
7291
7292 036254 032703 040000
7293 036260 001406
7294
7295 036262 012701 014506
7296 036266
      036266 104455
      036270 005677
      036272 014223
      036274 016136

7297
7298
7299
7300 036276 004767 164477
7301 036302 004767 163704

;REPORT "INTERRUPT BR LEVEL TEST ERROR:"
20$: ERRDF 3005,EM3001,ER0503; >>>> ERROR 03005 <<<<.
                                TRAP C$ERRDF
                                .WORD 3005
                                .WORD EM3001
                                .WORD ER0503

; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
;
; BIT 0BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
; BEQ 26$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
;
; BR 24$ ;SKIP THE BR CHECK IF NO RX INT OCCURRED.
;
; VERIFY THAT THE RX INTERRUPT WAS AT THE PROPER BR LEVEL.
;
22$: MOV R3,R4 ;CALCULATE THE BR LEVEL.
      BIC 0177400,R4 ; THAT THE RECEIVE
      ASR R4 ; INTERRUPT WAS
      ASR R4 ; REQUESTED AT, WHICH
      ASR R4 ; IS ONE GREATER THAN
      ASR R4 ; THE PROCESSOR PRIORITY
      ASR R4 ; LEVEL AT WHICH THE
      INC R4 ; RECEIVE INTERRUPT OCCURRED.
      MOV8 BRLEVL,R5 ;GET THE EXPECTED INTERRUPT BR LEVEL.
      CMPB R4,R5 ;COMPARE THE INTERRUPT BR LEVEL WITH EXPECTED.
      BEQ 24$ ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
;REPORT "RX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
      MOV 0EM3004,R1 ;SELECT ERROR MESSAGE FOR THE ERROR CALL.
      ERRDF 3006,EM3001,ER3001; >>>> ERROR 03006 <<<<.
                                TRAP C$ERRDF
                                .WORD 3006
                                .WORD EM3001
                                .WORD ER3001

; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
;
; BIT 0BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
; BEQ 26$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
;
; TEST FOR INTERRUPTS OCCURING IN THE PROPER ORDER.
;
24$: BIT 0BIT14,R3 ;CHECK THE IMPROPER INT ORDER ERROR FLAG.
      BEQ 26$ ;SKIP ERROR REPORT IF ERROR DID NOT OCCUR.
;REPORT "TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS RX INT."
      MOV 0EM3005,R1 ;SELECT THE ERROR MESSAGE FOR INDIRECT PRINT.
      ERRDF 3007,EM3001,ER0503; >>>> ERROR 03007 <<<<.
                                TRAP C$ERRDF
                                .WORD 3007
                                .WORD EM3001
                                .WORD ER0503

; CLEAN UP, EXIT THE TEST.
;
26$: JSR PC,IXIEO ;CLEAR TRANSMITTER INTERRUPTS.
      JSR PC,RXIEO ;CLEAR RECEIVER INTERRUPTS.

```

ADDRESS	HEX	ASCII	OPERATION	COMMENT
7302	036306	005067	143774	60\$:
7303	036312			CLR CTRC
	036312	012700	000340	SETPRI #PRI07
	036316	104441		
7304	036320			ENDIST
	036320			
	036320	104401		
7305				

```

7307 .SBIH HARDWARE TEST - DIABMP
7308 :. *****
7309 :* DIAGNOSTIC FIELD (BMP) TEST :
7310 :* THIS TEST VERIFIES THAT A REQUEST TO THE DUT TO REPORT BMP STATUS
7311 :* CODES IS COMPLIED WITH, WITHIN THE SPECIFIED TIME.
7312 :* ALL ACTIVE LINES ARE TESTED.
7313 :. *****
7314
7315 036322 BGNIST
7316 036322 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T25::
7317 036322 012700 000240 MOV #PRI05,RO
7318 036322 104441 TRAP C1SPRI
7319 036322 000031 INUM == INUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
7320 036330 012767 000031 143766 MOV #INUM,TSTNUM ;SET UP THE TEST NUMBER. (31)
7321 036336 012767 177777 143742 MOV #1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
7322 036344 012767 000001 145414 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
7323 036352 012767 006035 145410 MOV #3101,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
7324 036360 012767 014600 145404 MOV #EM3101,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRtbl.
7325 036366 012767 017222 145400 MOV #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
7326
7327 :. RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
7328 :. CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
7329 :. THIS SUBROUTINE REPORTS ERROR >>>> 3101 <<<<.
7330 036374 004767 161402 JSR PC,CLRST ;RESET THE DUT-11, REPORT ANY ERRORS FOUND.
7331 036400 103103 BCC 60$ ;RESET FAILURE?, ABORT THIS TEST.
7332
7333 :. TEST ALL ACTIVE LINES INDIVIDUALLY.
7334 :. WRITE THE REQUEST CODE TO THE DIAGNOSTIC FIELD IN THE LPR REGISTER.
7335 :. VERIFY THAT A BMP CODE IS RETURNED WITHIN THE CORRECT TIME.
7336 036402 016705 143622 MOV ACTINS,R5 ;GET THE ACTIVE LINE BIT MAP.
7337 036406 005004 CLR R4 ;CLEAR THE LINE NUMBER COUNTER.
7338 036410 016703 143626 MOV CSRA,R3 ;GET THE ADDRESS OF THE DUT'S CSR.
7339 036414 000241 2$ CLR C ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
7340 036416 006005 ROR R5 ;SHIFT THE BIT MAP INTO THE CARRY BIT.
7341 036420 103070 BCC 9$ ;DO NOT TEST THE LINE IF IT IS INACTIVE.
7342
7343 :. SELECT THE LINE UNDER TEST.
7344 :. WRITE THE BMP REQUEST CODE TO THE DIAG FIELD IN THE LPR REGISTER.
7345
7346 036422 012767 006036 145340 MOV #3102,ERRNBR ;SET THE ERROR NUMBER TO 3102.
7347 036430 010413 MOV R4,(R5) ;SELECT THE LINE CURRENTLY UNDER TEST.
7348 036432 052777 000002 143606 BIS #2,LPR ;WRITE THE BMP REQUEST CODE TO THE LPR.
7349
7350 :. WAIT FOR BMP REQUEST CODE TO BE CLEARED, REPORT ERROR IF TIME-OUT
7351 :. OCCURS.
7352
7353 036440 012701 011750 MOV #11750,R1 ;TEST BIT 1, TIMEOUT OF 1 SEC.
7354 036444 016702 143576 MOV LPR,R2 ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7355 036450 004767 164540 JSR PC,WAITBIC ;WAIT FOR REQUEST CODE TO CLEAR.
7356 036454 103042 BCC 6$ ;GO REPORT ERROR IF CODE DID NOT CLEAR IN TIME.
7357
7358 :. WAIT FOR BMP CODE TO APPEAR IN THE FIFO, REPORT ERROR IF TIME OUT
7359 :. OCCURS.
7360

```

```

7361 036456 005267 145306      INC      ERRNBR      ;SET ERROR NUMBER TO 3103.
7362 036462 012701 070012      MOV      #70012,R1    ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
7363 036466 016702 143550      MOV      CSRA,R2    ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7364 036472 004767 164572      JSR      PC,WAIBIS   ;WAIT FOR RX DATA AVAILABLE TO SET.
7365 036476 105031              BCC      6$          ;GO REPORT ERROR IF CODE DID NOT CLEAR IN TIME.
7366                          ;*
7367                          ; READ THE BMP CODE (IF IT IS THERE) FROM THE RBUF REGISTER.
7368                          ; DETERMINE IF IT IS A VALID BMP CODE.
7369                          ; VERIFY THE BMP CODE WAS RECEIVED FROM THE CORRECT CHANNEL.
7370                          ; IF THE BMP CODE DOES NOT INDICATE OUT RUNNING OK, THEN SAVE IT ON
7371                          ; THE QUEUE TO BE REPORTED IN A LATER TEST.
7372                          ;*
7373 036500 005267 145264      INC      ERRNBR      ;SET ERROR NUMBER TO 3104.
7374 036504 017702 143534      MOV      @RBUF,R2    ;GET THE BMP CODE FROM THE FIFO.
7375 036510 100024              BPL      6$          ;GO REPORT ERROR IF NO BMP CODE FOUND.
7376 036512 005267 145252      INC      ERRNBR      ;SET ERROR NUMBER TO 3105.
7377 036516 012700 170301      MOV      #170301,R0    ;SET-UP A BMP CODE MASK.
7378 036522 040200              BIC      R2,R0        ;TRY TO CLEAR THE BMP MASK.
7379 036524 001016              BNE      6$          ;GO REPORT ERROR IF IT IS NOT A VALID BMP CODE.
7380 036526 005267 145236      INC      ERRNBR      ;SET THE ERROR NUMBER TO 3106.
7381 036532 010200              MOV      R2,R0        ;COPY THE BMP CODE.
7382 036534 000300              SWAB     R0          ;PUT THE LINE NUMBER IN THE LOW BYTE.
7383 036536 042700 177760      BIC      #177760,R0    ;CLEAR THE UNWANTED BITS.
7384 036542 120400              CMPB     R4,R0        ;DID THE BMP CODE COME FROM THE CORRECT LINE?.
7385 036544 001006              BNE      6$          ;NO; GO REPORT ERROR.
7386 036546 120227 000305      CMPB     R2,#305    ;IS THE BMP CODE A "GOOD ONE"?.
7387 036552 001413              BEQ      8$          ;YES; SKIP SAVING THE BMP CODE ON THE QUEUE.
7388 036554 004767 163516      JSR      PC,SAVBMP   ;SAVE THE BMP CODE ON THE QUEUE.
7389 036560 000410              BR       8$          ;GO SEE IF THERE ARE ANY MORE LINE TO TEST.
7390
7391 036562 010401              6$:      MOV      R4,R1        ;PASS THE LINE NUMBER TO BE REPORTED.
7392 036564 012702 014643      MOV      @EM3102,R2    ;PASS THE ERROR MESSAGE TO BE REPORTED.
7393                                ; "BMP REQUEST BIT BAD ON LINE:"
7394 036570              ERROR                                ;
7395 036570 104460              TRAP      C$ERROR
7396
7397                          ;*
7398                          ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7399 036572 052767 000100 143424      BIT      @BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
7400 036600 001403              BEQ      60$          ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7401
7402                          ;*
7403                          ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
7404                          ;
7405 036602 005204      8$:      INC      R4          ;INCREMENT THE LINE NUMBER COUNTER.
7406 036604 005205              TEST     R4          ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
7407 036606 001302              BNE      6$          ;YES; BRANCH TO TEST THE NEXT LINE.
7408 036610 005067 143472      60$:      CLR      CTRLCH    ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7409 036614              ENDTST
7409 036614 104401              L10056:      TRAP      C$TEST

```

```

7411
7412          .TITLE  HARDWARE TEST          - REPBMF -
7413          ;* *****
7414          ;*          REPORT ANY BMP CODES IN THE QUEUE
7415          ;*          THIS IS A PSEUDO-TEST USED TO REPORT ANY BMP CODES THAT WERE FOUND
7416          ;*          IN THE DUT'S FIFO DURING PREVIOUS TEST, AND LOGGED IN THE BMP CODE
7417          ;*          QUEUE.
7418          ;*          IT IS UNLIKELY THAT RUNNING THIS PSEUDO-TEST ALONE WILL PRODUCE ANY
7419          ;*          ERROR REPORTS.
7420          ;*
7421          ;* *****
7422          BGNTST
7423          TNUM ** TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
7424          MOV  *TNUM,TSTNUM          ;SET UP THE TEST NUMBER. (93)
7425          MOV  *-1,CTRLCF          ;INDICATE THAT WE ARE IN A TEST.
7426          MOV  BMPCQP,R2          ;GET THE CONTENTS OF THE POINTER.
7427          MOV  *BMPCQB,R3          ;GET THE START ADDRESS OF THE QUEUE.
7428          CMP  R2,R3          ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
7429          BEQ  60$          ;EXIT NO CODES IN THE QUEUE.
7430          ;*
7431          ;* THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
7432          ;*
7433          ;* REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN"
7434          MOV  *EM9304,R1          ;PASS THE FIRST MESSAGE TO BE REPORTED.
7435          ERDF  9301,EM9301,ER9301 ;>>>> ERROR 09301 <<<<<.
7436          036652 104455          TRAP  C$ERDF
7437          036654 022125          .WORD  9301
7438          036656 015257          .WORD  EM9301
7439          036660 017262          .WORD  ER9301
7440          MOV  *BMPCQB,BMPCQP          ;SET POINTER BACK TO THE BEGINING OF THE QUE.
7441          CLR  CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7442          ENDTST
7443          L10057: TRAP  C$ETST
7444          036674 104401

```



```

7450
7451
7452
7453
7454
7455
7456
7457
7458
7459
7460
7461
7462
7463
7464
7465 036676      BGNHRD
      036676      000022
      036700
7466
7476
7477 036700      ;DEVICE CSR ADDRESS QUESTION:
      036700      000031      GPRMA HWPTQ1,0,0,160000,177776,YES
      036702      036744
      036704      160000
      036706      177776
7478
7479 036710      ;DEVICE INTERRUPT VECTOR QUESTION:
      036710      001031      GPRMA HWPTQ2,2,0,40,776,YES
      036712      036762
      036714      000040
      036716      000776
7480
7481 036720      ;ACTIVE LINES BIT MAP QUESTION:
      036720      002032      GPRMD HWPTQ3,4,0,MAPLNS,0,MAPLNS,YES
      036722      037015
      036724      177777
      036726      000000
      036730      177777
7482
7483 036732      ;INTERRUPT BR LEVEL QUESTION:
      036732      003032      GPRMD HWPTQ4,6,0,377,0,6,YES
      036734      037043
      036736      000377
      036740      000000
      036742      000006
7484
7485
7486 036744      ENDHRD
      036744
7487
7494
7495 036744      103      123      122
      036747      040      101      104
      036752      104      122      105
      036755      123      123      072
      036760      040      000

```

.SBTTL HARDWARE PARAMETER CODING SECTION
 ;
 ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 ; WITH THE OPERATOR.
 ;
 ;--
 L\$HARD: .WORD L10060-L\$HARD/2
 L\$HARD: .WORD T\$CODE
 .WORD HWPTQ1
 .WORD T\$LOULIM
 .WORD T\$HILIM
 .WORD T\$CODE
 .WORD HWPTQ2
 .WORD T\$LOULIM
 .WORD T\$HILIM
 .WORD T\$CODE
 .WORD HWPTQ3
 .WORD MAPLNS
 .WORD T\$LOULIM
 .WORD T\$HILIM
 .WORD T\$CODE
 .WORD HWPTQ4
 .WORD 377
 .WORD T\$LOULIM
 .WORD T\$HILIM
 L10060: .EVEN
 L10060:
 HWPTQ1: .ASCII /CSR ADDRESS: /

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


```

7509
7510
7511      .SBTTL  SOFTWARE PARAMETER CODING SECTION
7512
7513      ;**
7514      ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
7515      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7516      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7517      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
7518      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
7519      ; WITH THE OPERATOR.
7520      ;**
7521
7522      BGNSFT
7523      037070      000017      .WORD  L10061-L$SOFT/2
7524      037070      000017      L$SOFT::
7525      037072
7526      ;UNIT NUMBER PRINTOUT QUESTION:
7527      037072      000130      GPRML  SWPTQ1,0,20,YES
7528      037074      037130      .WORD  T$CODE
7529      037076      000020      .WORD  SWPTQ1
7530      .WORD  20
7531
7532      ;ROM VERSION NUMBER PRINTOUT ON FIRST PASS QUESTION:
7533      037100      000130      GPRML  SWPTQ2,0,1,YES
7534      037102      037204      .WORD  T$CODE
7535      037104      000001      .WORD  SWPTQ2
7536      .WORD  1
7537
7538      ;EXTENDED ERROR REPORTING QUESTION:
7539      037106      000130      GPRML  SWPTQ3,0,100,YES
7540      037110      037255      .WORD  T$CODE
7541      037112      000100      .WORD  SWPTQ3
7542      .WORD  100
7543
7544      ; IF EXTENDED ERROR REPORTING IS NOT REQUIRED THEN SKIP THE NEXT QUESTION.
7545      ;**
7546      XFERF  ENDD
7547      037114      006044      .WORD  T$CODE
7548
7549      ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
7550      037116      001052      GPRMD  SWPTQ4,2,D,177777,0,177777,YES
7551      037120      037310      .WORD  T$CODE
7552      037122      177777      .WORD  SWPTQ4
7553      037124      000000      .WORD  177777
7554      037126      177777      .WORD  T$LOLIM
7555      .WORD  T$HILIM
7556
7557      .EVEN
7558
7559      ENDD:  ENDSFT
7560
7561      .EVEN
7562      L10061:
7563
7564      SWPTQ1:  .ASCIIZ  /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /
7565      037130      122      105      120
7566      037133      117      122      124
7567      037136      040      125      116
7568      037141      111      124      040
7569      037144      116      125      115

```

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

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

ACTINS	002230 G	C1AU	000052	DROP	025122	EM902	010766 G	EM9023	015217 G
ADR	000020 G	C1AUTO	000061	DR00MG	005464 G	EM1001	011027 G	EM9024	015241 G
ADRPTR	017764 G	C1BRK	000022	DR02MG	005470 G	EM1002	011055 G	EM9301	015257 G
ALTFLD	017454 G	C1BSEG	000004	DR04MG	005475 G	EM1003	011142 G	EM9302	015336 G
ASSEMB	000010	C1BSUB	000002	DR06MG	005501 G	EM1101	011220 G	EM9303	015366 G
BCOUNT	002352 G	C1CEFG	000045	DR10MG	005517 G	EM1201	011273 G	EM9304	015433 G
BITTBL	002410 G	C1CLCK	000062	DR12MG	005526 G	EM1202	011321 G	ENDD	037130
BIT0	000001 G	C1CLEA	000012	DR14MG	005537 G	EM1203	011405 G	ENDFTB	003726 G
BIT00	000001 G	C1CLOS	000035	DR16MG	005550 G	EM1204	011463 G	ENDIT	025042
BIT01	000002 G	C1CLP1	000006	EDROP	025200	EM1205	011517 G	ERCNTB	002464 G
BIT02	000004 G	C1CVEC	000036	EF.CON	000036 G	EM1301	011543 G	ERLTBI	002726 G
BIT03	000010 G	C1DCIN	000044	EF.NEW	000035 G	EM1302	011576 G	ERRBLK	003774 G
BIT04	000020 G	C1DOOU	000051	EF.PWR	000034 G	EM1401	011635 G	ERRMG	003772 G
BIT05	000040 G	C1DRPT	000024	EF.RES	000037 G	EM1402	011674 G	ERRNBR	003770 G
BIT06	000100 G	C1DU	000053	EF.STA	000040 G	EM1403	011762 G	ERRTYP	003766 G
BIT07	000200 G	C1EDIT	000003	EF0503	004074 G	EM1404	012035 G	ERSMRF	002462 G
BIT08	000400 G	C1ERDF	000055	EF0505	004101 G	EM1405	012107 G	EP0101	015510 G
BIT09	001000 G	C1ERRR	000056	EF1401	004154 G	EM1406	012122 G	ERG001	016042 G
BIT1	000002 G	C1ERR0	000060	EF1402	004256 G	EM1407	012135 G	ER0503	016136 G
BIT10	002000 G	C1ERSF	000054	EF1601	004313 G	EM1408	012147 G	ER0504	016174 G
BIT11	004000 G	C1ERS0	000057	EF1602	004345 G	EM1501	012155 G	ER1401	015254 G
BIT12	010000 G	C1ESCA	000010	EF1603	004407 G	EM1502	012203 G	ER1601	016416 G
BIT13	020000 G	C1ESEG	000005	EF1604	004451 G	EM1601	012221 G	ER1603	016526 G
BIT14	040000 G	C1ESUB	000003	EF3001	004546 G	EM1604	012304 G	ER3001	016620 G
BIT15	100000 G	C1ETST	000001	EF3002	004615 G	EM1701	012360 G	ER9004	016722 G
BIT2	000004 G	C1EXIT	000032	EF9006	004664 G	EM1801	012443 G	ER9007	017036 G
BIT3	000010 G	C1GETB	000026	EF9010	004710 G	EM1901	012517 G	ER9008	017126 G
BIT4	000020 G	C1GETW	000027	EF9016	004777 G	EM2001	012602 G	ER9101	017222 G
BIT5	000040 G	C1GMAN	000043	EF9017	005074 G	EM2002	012652 G	ER9301	017262 G
BIT6	000100 G	C1GPHR	000042	EF9018	005150 G	EM2301	012725 G	EVL	000004 G
BIT7	000200 G	C1GPLD	000030	EF9301	005255 G	EM2302	012772 G	EXDERR	002304 G
BIT8	000400 G	C1GPR1	000040	EF9302	005333 G	EM2401	013030 G	EXEND	002100
BIT9	001000 G	C1INIT	000011	EM0101	020360 G	EM2601	013073 G	EXLOAD	000035
BMPQOB	002526 G	C1INLP	000020	EM0102	020444 G	EM2602	013131 G	FDATA	002250 G
BMPQOF	002726 G	C1MANI	000050	EM0103	005560 G	EM2603	013222 G	FLISA	002250 G
BMPQOP	002524 G	C1MEM	000031	EM0201	005616 G	EM2604	013310 G	FLSO	000006 G
BOE	000400 G	C1MSG	000023	EM0202	005651 G	EM2605	013404 G	FLAU	000015
BRLEV	002240 G	C1OPEN	000034	EM0203	006024 G	EM2606	013467 G	FLAUTO	000020
BUF BAS	002726 G	C1PNTB	000014	EM0204	006167 G	EM2607	013526 G	FLBGN	000040
BUFEND	003726 G	C1PNTF	000017	EM0301	006346 G	EM2608	013622 G	FLCLEA	000007
BUFMID	003326 G	C1PNTS	000016	EM0302	006420 G	EM2609	013673 G	FLDU	000016
BUFPTR	002302 G	C1PNTX	000015	EM0303	006560 G	EM2610	013752 G	FLEND	000041
BUFSQT	003526 G	C1QIO	000377	EM0401	006737 G	EM2611	014044 G	FLHARD	000004
CACHRX	023632 G	C1RDBU	000007	EM0402	007015 G	EM2612	014127 G	FLHW	000013
CACHTX	023660 G	C1REFG	000047	EM0501	007165 G	EM3001	014223 G	FLINIT	000006
CALHSI	017526 G	C1RESE	000033	EM0502	007241 G	EM3002	014262 G	FLJMP	000050
CMTRAP	017752 G	C1REVI	000003	EM0503	007415 G	EM3003	014356 G	FLMOD	000000
CLKBRL	002336 G	C1RFLA	000021	EM0504	007505 G	EM3004	014432 G	FLMSG	000011
CLKCSR	002334 G	C1RPT	000025	EM060	007575 G	EM3005	014506 G	FLPROT	000021
CLKHRZ	002342 G	C1SEFG	000046	EM0601	007637 G	EM3101	014600 G	FLPWR	000017
CLKINT	023706 G	C1SPRI	000041	EM0602	010017 G	EM3102	014643 G	FLRPT	000012
CLKVEC	002340 G	C1SVEC	000037	EM0701	010202 G	EM9014	014722 G	FLSEG	000003
CLNRST	020002 G	C1TPRI	000013	EM0702	010245 G	EM9017	015016 G	FLSOFT	000005
CLR16W	020024 G	DELAT	020124 G	EM0703	010425 G	EM9018	015127 G	FLSRV	000010
CNTRR	020046 G	DEPTBL	002212 G	EM0801	010610 G	EM9019	015137 G	FLSRB	000007
CSRA	002242 G	DIAGMC	000000	EM0802	010656 G	EM9020	015154 G	FLSW	000014
CTRLCF	002306 G	DRADRT	002242 G	EM0901	010726 G	EM9022	015200 G	FLTEST	000001

GETPRM 024640	LPRO 000004 G	L10002 015624	MSLOUP 020300 G	SPTBL 002224 G
GPRSOB 002450 G	L\$ACP 002110 G	L10003 016134	MSTICK 002354 G	SPPSTS 022344 G
G\$CNTD 000200	L\$APT 002036 G	L10004 016172	NDERPT 002226 G	SVCGBL 000000
G\$DELM 000372	L\$AU 025206 G	L10005 016252	NEWPAS 024620	SVCINS 000001
G\$DISP 000003	L\$AUT 002070 G	L10006 016414	NEWRES 024612	SVCUSX 000001
G\$EXCP 000400	L\$AUTO 025056 G	L10007 016524	NEWSTA 024302	SVCTAG 000001
G\$HILI 000002	L\$CCP 002106 G	L10010 016616	NUMLNS 000020 G	SVCTST 000001
G\$LOLI 000001	L\$CLEA 025060 G	L10011 016720	OOPS 020314 G	SWAPO 022422 G
G\$NO 000000	L\$CO 002032 G	L10012 017034	OPTION 002224 G	SWPTQ1 037130
G\$OFFS 000400	L\$DEPO 002011 G	L10013 017124	O\$APTS 000000	SWPTQ2 037204
G\$OFST 000376	L\$DESC 004046 G	L10014 017220	O\$AU 000000	SWPTQ3 037255
G\$PRMA 000001	L\$DESP 002076 G	L10015 017230	O\$BGNR 000001	SWPTQ4 037310
G\$PRMD 000002	L\$DEVP 002060 G	L10016 017452	O\$BGNB 000001	SLSYM 010000
G\$PRM 000000	L\$DISP 002124 G	L10017 024226	O\$DU 000001	TIMER1 002344 G
G\$RADA 000140	L\$DLY 002116 G	L10021 025055	O\$ERRT 000001	TIMER2 002346 G
G\$RADB 000000	L\$DTP 002040 G	L10022 025057	O\$GNSW 000001	TIMER3 002350 G
G\$RADD 000040	L\$DTYP 002034 G	L10023 025074	O\$POIN 000001	TNUM 000032 G
G\$RADL 000120	L\$DU 025076 G	L10024 025204	O\$SETU 000000	TP4FLG 002320 G
G\$RADO 000020	L\$DUT 002072 G	L10025 025212	PARQA 002366 G	TP4RTN 024124 G
G\$XFER 000004	L\$DVTY 004036 G	L10026 025474	PASCNT 002312 G	TP4VEC 002322 G
G\$YES 000010	L\$EF 002052 G	L10027 025724	PCSLDT 000016 G	TSABRT 022472 G
HELP 000000	L\$ENVT 002044 G	L10030 026170	PNT 001000 G	TSTNUM 002324 G
HOL 000000 G	L\$ERRT 003766 G	L10031 026366	PREGRT 004020 G	TXAD1A 002254 G
HWPTQ1 036744	L\$ETP 002102 G	L10032 026560	PREGOS 003776	TXAD2A 002256 G
HWPTQ2 036762	L\$EXP1 002046 G	L10033 025776	PRI 002000 G	TXBFCA 002260 G
HWPTQ3 037015	L\$EXP4 002064 G	L10034 027204	PRI00 000000 G	TXBFCO 000016 G
HWPTQ4 037043	L\$EXP5 002066 G	L10035 027412	PRI01 000040 G	TXDSBL 022604 G
IHE 010000 G	L\$HARD 036700 G	L10036 027606	PRI02 000100 G	TXENBL 022700 G
IDU 000040 G	L\$HIME 002120 G	L10037 030022	PRI03 000140 G	TXIEO 022774 G
IER 020000 G	L\$HPCP 002016 G	L10040 030250	PRI04 000200 G	TXIE1 023034 G
ILESTAT 002310 G	L\$HPTP 002022 G	L10041 030544	PRI05 000240 G	TXINTC 002326 G
ISR 000100 G	L\$HW 002212 G	L10042 031032	PRI06 000300 G	TXINTF 002330 G
IXE 004000 G	L\$ICP 002104 G	L10043 031432	PRI07 000340 G	TXINTR 024146 G
I\$AU 000041	L\$INIT 024236 G	L10044 031774	PUFIFO 020542 G	TXVECA 002234 G
I\$AUTO 000041	L\$IADP 002026 G	L10045 032242	RBUFA 002244 G	T\$ARGC 000003
I\$CLN 000041	L\$LAST 037454 G	L10046 032432	RDPDR 020624 G	T\$CODE 001052
I\$DU 000041	L\$LOAD 002100 G	L10047 032750	REGTST 021076 G	T\$ERRN 022125
I\$HRD 000041	L\$LUN 002074 G	L10050 033212	REPSMR 021330 G	T\$EXCP 000000
I\$INIT 000041	L\$MREV 002050 G	L10051 033326	RESETT 021356 G	T\$FLAG 000050
I\$MOD 000041	L\$NAME 002000 G	L10052 033644	RHATBB 002262 G	T\$GMAN 000000
I\$MSG 000041	L\$PRIO 002042 G	L10053 034220	ROLDAP 021470 G	T\$HILI 177777
I\$PROT 000040	L\$PROT 024230 G	L10054 035356	RSTRTT 021514 G	T\$LAST 000001
I\$PTAB 000041	L\$PRT 002112 G	L10055 036320	RXBRTT 023756 G	T\$LOLI 000000
I\$PWR 000041	L\$REPP 002062 G	L10056 036614	RXIEO 022212 G	T\$LSTM 010000
I\$RPT 000041	L\$REV 002010 G	L10057 036674	RXIE1 022252 G	T\$LTNO 000032
I\$SEG 000041	L\$RPT 024222 G	L10060 036744	RXINPT 024036 G	T\$NEST 177777
I\$SETU 000041	L\$SOFT 037072 G	L10061 037130	RXINTC 002314 G	T\$NSO 000000
I\$SET 000041	L\$SPC 002056 G	HAPLNS 177777 G	RXINTF 002316 G	T\$NS1 000005
I\$SRV 000041	L\$SPCP 002020 G	M\$UNIT 005433 G	RXIMA 002244 G	T\$PINA 000000
I\$SUB 000041	L\$STPT 002024 G	M\$ENAB 002364 G	RXVECA 002232 G	T\$SAVL 177777
I\$TST 000041	L\$STA 002030 G	M\$MPRES 002362 G	ROSLDT 000002 G	T\$SEGL 177777
J\$IMP 000167	L\$SW 002224 G	M\$MSRO 002360 G	RSLDT 000004 G	T\$SUBN 000000
INCTRA 002252 G	L\$TEST 002114 G	MSG1 015626 G	RSLDT 000006 G	T\$TAGI 177777
IOE 040000 G	L\$TIML 002014 G	MSG2 015704 G	RSLDT 000010 G	T\$TAGN 010062
IOI 000010 G	L\$UNIT 002012 G	MSG3 015763 G	RSLDT 000012 G	T\$TEMP 000000
IPCULT 000036 G	L10000 002222	MSLCNT 002356 G	RSLDT 000014 G	T\$TEST 000032
LPRA 002346 G	L10001 002230	MSLGET 020164 G	SAVBMP 022276 G	T\$TSTM 177777

D16

CZDHUBO.DM11 FUNC TST PART1 MACRO M1200 15 MAR 84 09:15 PAGE 128 3

SEQ 198

T11STIS= 000001	T11SOF= 010061	T117 032244 G	T15 025726 G	WAIBIC 023214 G
T11SAU= 010025	T11SW= 010001	T118 032434 G	T14 026172 G	WAIBIS 023270 G
T11SAUT= 010022	T11TES= 010057	T119 032752 G	T13 026370 G	WDPOR 023344 G
T11CLE= 010023	T11 025214 G	T12 025476 G	T16 026562 G	WORD1 002332 G
T11DU= 010024	T110 027610 G	T120 033214 G	T17 027000 G	WTWLC 023552 G
T11HAR= 010060	T111 030024 G	T121 033330 G	T18 027206 G	WTWLP 023602 G
T11HW= 010000	T112 030252 G	T122 033646 G	T19 027414 G	X1ALWA= 000000
T11INI= 010021	T113 030546 G	T123 034222 G	UAM= 000200 G	X1FALS= 000040
T11MSG= 010016	T114 031034 G	T124 035360 G	UNBITB 002370 G	X1OFFS= 000400
T11PRO= 010020	T115 031434 G	T125 036322 G	UNITN 002236 G	X1TRUK= 000020
T11RPT= 010017	T116 031776 G	T126 036616 G	UNSDIV 023060 G	SPATCH 037400 G

ABS 032454 000
000000 001
ERRORS DETECTED: 0

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

PROGRAM DOCUMENT	B1	GLOBAL ERROR REPORTI....B5	GLOBAL SUBROUTINE	B9	HARDWARE TEST	B13
PROGRAM DOCUMENT	C1	GLOBAL ERROR REPORTI....C5	GLOBAL SUBROUTINE	C9	HARDWARE TEST	C13
PROGRAM DOCUMENT	D1	GLOBAL ERROR REPORTI....D5	GLOBAL SUBROUTINE	D9	HARDWARE TEST	D13
PROGRAM DOCUMENT	E1	GLOBAL ERROR REPORTI....E5	GLOBAL SUBROUTINE	E9	HARDWARE TEST	E13
PROGRAM DOCUMENT	F1	GLOBAL ERROR REPORTI....F5	GLOBAL SUBROUTINE	F9	HARDWARE TEST	F13
PROGRAM DOCUMENT	G1	GLOBAL ERROR REPORTI....G5	GLOBAL SUBROUTINE	G9	HARDWARE TEST	G13
PROGRAM DOCUMENT	H1	GLOBAL ERROR REPORTI....H5	GLOBAL SUBROUTINE	H9	HARDWARE TEST	H13
PROGRAM DOCUMENT	I1	GLOBAL ERROR REPORTI....I5	INTERRUPT SERVICE RO....I9	HARDWARE TEST	I13	
PROGRAM DOCUMENT	J1	GLOBAL ERROR REPORTI....J5	INTERRUPT SERVICE RO....J9	HARDWARE TEST	J13	
PROGRAM DOCUMENT	K1	GLOBAL SUBROUTINES S....K5	INTERRUPT SERVICE RO....K9	HARDWARE TEST	K13	
PROGRAM DOCUMENT	L1	GLOBAL SUBROUTINE	INTERUPT SERVICE ROU....L9	HARDWARE TEST	L13	
PROGRAM DOCUMENT	M1	GLOBAL SUBROUTINE	INTERUPT SERVICE ROU....M9	HARDWARE TEST	M13	
PROGRAM DOCUMENT	N1	GLOBAL SUBROUTINE	GLOBAL TRAP SERVICEN9	HARDWARE TEST	N13	
PROGRAM DOCUMENT	B2	GLOBAL SUBROUTINE	INTERUPT SERVICE ROU....B10	HARDWARE TEST	B14	
PROGRAM DOCUMENT	C2	GLOBAL SUBROUTINE	INTERUPT SERVICE ROU....C10	HARDWARE TEST	C14	
PROGRAM DOCUMENT	D2	GLOBAL SUBROUTINE	PROTECTION TABLE	HARDWARE TEST	D14	
PROGRAM DOCUMENT	E2	GLOBAL SUBROUTINE	PROTECTION TABLE	HARDWARE TEST	E14	
PROGRAM DOCUMENT	F2	GLOBAL SUBROUTINE	INITIALIZE SECTION	HARDWARE TEST	F14	
PROGRAM DOCUMENT	G2	GLOBAL SUBROUTINE	INITIALIZE SECTION	HARDWARE TEST	G14	
PROGRAM DOCUMENT	H2	GLOBAL SUBROUTINE	INITIALIZE SECTION	HARDWARE TEST	H14	
PROGRAM DOCUMENT	I2	GLOBAL SUBROUTINE	INITIALIZE SECTION	HARDWARE TEST	I14	
PROGRAM DOCUMENT	J2	GLOBAL SUBROUTINE	INITIALIZE SECTION	HARDWARE TEST	J14	
PROGRAM DOCUMENT	K2	GLOBAL SUBROUTINE	AUTOCROP SECTION	HARDWARE TEST	K14	
PROGRAM DOCUMENT	L2	GLOBAL SUBROUTINE	CLEANUP CODING SECTI....K10	HARDWARE TEST	L14	
PROGRAM HEADER	M2	GLOBAL SUBROUTINE	DROP UNIT SECTION	HARDWARE TEST	M14	
PROGRAM HEADER	N2	GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE TEST	N14	
DISPATCH TABLE	B3	GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE TEST	B15	
DISPATCH TABLE	C3	GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE TEST	C15	
DEFAULT HARDWARE P-T....D3		GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE TEST	D15	
SOFTWARE P-TABLE	E3	GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE TEST	E15	
GLOBAL EQUATES SECTI....F3		GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE TEST	F15	
GLOBAL EQUATES SECTI....G3		GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE TEST	G15	
GLOBAL DATA SECTIONH3		GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE TEST	H15	
GLOBAL DATA SECTIONI3		GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE TEST	I15	
GPR HANDLING ROUTINE....J3		GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE TEST	J15	
GPR FRAME ACCESS EQU....K3		GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE PARAMETER C....K15		
GLOBAL MACRO DEFINIT....L3		GLOBAL SUBROUTINE	HARDWARE TEST	HARDWARE PARAMETER C....L15		
GLOBAL MACRO DEFINIT....M3		GLOBAL SUBROUTINE	HARDWARE TEST	SOFTWARE PARAMETER C....M15		
GLOBAL SUBROUTINE	N3	GLOBAL SUBROUTINE	HARDWARE TEST	SOFTWARE PARAMETER C....N15		
GLOBAL TEXT SECTIONB4		GLOBAL SUBROUTINE	HARDWARE TEST	SYMBOL TABLE	B16	
GLOBAL TEXT SECTIONC4		GLOBAL SUBROUTINE	HARDWARE TEST	SYMBOL TABLE	C16	
GLOBAL TEXT SECTIOND4		GLOBAL SUBROUTINE	HARDWARE TEST	SYMBOL TABLE	D16	
GLOBAL TEXT SECTIONE4		GLOBAL SUBROUTINE	HARDWARE TEST			
GLOBAL TEXT SECTIONF4		GLOBAL SUBROUTINE	HARDWARE TEST			
GLOBAL ERROR REPORTI....G4		GLOBAL SUBROUTINE	HARDWARE TEST			
GLOBAL ERROR REPORTI....H4		GLOBAL SUBROUTINE	HARDWARE TEST			
GLOBAL ERROR REPORTI....I4		GLOBAL SUBROUTINE	HARDWARE TEST			
GLOBAL ERROR REPORTI....J4		GLOBAL SUBROUTINE	HARDWARE TEST			
GLOBAL ERROR REPORTI....K4		GLOBAL SUBROUTINE	HARDWARE TEST			
GLOBAL ERROR REPORTI....L4		GLOBAL SUBROUTINE	HARDWARE TEST			
GLOBAL ERROR REPORTI....M4		GLOBAL SUBROUTINE	HARDWARE TEST			
GLOBAL ERROR REPORTI....N4		GLOBAL SUBROUTINE	HARDWARE TEST			