

B1

CZDHVBO DHU-11 FUNC TST PART2 MACRO M1200 15-MAR 84 09:28 PAGE 2
PROGRAM DOCUMENT

SEQ 1

.REN &

IDENTIFICATION

PRODUCT CODE: AC-T796B-MC
PRODUCT NAME: CZDHVBO DHU-11 FUNC TST PART2
PRODUCT DATE: 3 MARCH 1984
MAINTAINER: FNE - 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 ON 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

POP
DECUS

UNIBUS
DECTAPE

MASSBUS

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

ORIGINAL RELEASE:	15 DEC-83	ANTHONY HART
VERSION B0	3-MAR-84	ANTHONY HART

THE FOLLOWING MODIFICATIONS HAVE BEEN MADE TO THE OLD CZDHVA:

THE HARDWARE QUESTION "TYPE OF LOOPBACK" HAS BEEN ALTERED TO INCLUDE THE STAGGERED LOOPBACK CONNECTORS ON THE DHU11 DISTRIBUTION PANEL (H3029).

THE HARDWARE QUESTION "BR LEVEL" HAS BEEN REMOVED.

THE MODEM SIGNAL TESTS THAT WERE IN THE OLD VERSION (CZDHVA0) HAVE NOW BEEN REMOVED TO PART CZDHW. THEY WERE TESTS: 16 THRU 23, IN CZDHVA0.

THE FOLLOWING NEW TESTS HAVE BEEN ADDED TO THIS PART:

- TEST 13 - RXTIMER REGISTER TEST
- TEST 14 - TX_ACTION FIFO TEST
- TEST 15 - TX_FIFO 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
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

CZDHVBO IS PART OF THE DHU-11 FUNCTIONAL VERIFICATION TEST. THIS PART OF THE TEST VERIFIES THAT THE MAJOR COMMUNICATIONS FUNCTIONS OF THE BOARD ARE FUNCTIONING CORRECTLY. THIS PROGRAM DOES NOT PERFORM EXTENSIVE DATA TRANSMISSION AND RECEPTION TESTS.

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

1.2 SYSTEM REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DHU FVT:

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

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE PROCESSOR, THE UNIBUS, THE SYSTEM MEMORY, THE CONSOLE TERMINAL AND THE I/O 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 "DDDD".

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:DDDD	EXECUTE DDDDD PASSES (DDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. SEE THE FLAGS SECTION OF THIS DOCUMENT.
/EOP:DDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDD = 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 FLAG SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

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

*SEE THE ERROR INFORMATION SECTION OF THIS DOCUMENT.

SEE THE XXOP 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:

ZFLAGS:LOE:IER:BOE

2.4 EXTENDED COMMAND SYNTAX

2.4.1 START COMMAND

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

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

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

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

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

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

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

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED.
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR.
IER	INHIBIT ERROR REPORTING.
IBE	INHIBIT BASIC ERROR REPORTS.
IXF	INHIBIT EXTENDED ERROR REPORTS.
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER.
PNT	PRINT NUMBER OF TEST BEING EXECUTED.
BOE	BELL ON ERROR (NOT RELATED TO BELL PROMPTING).
UAM	RUN IN UNATTENDED MODE, REPAIRING 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 (/EGP:<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:IOE=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.

K1

2.4.2 RESTART COMMAND -

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

2.4.2.1 TESTS, PASS, AND FLAGS SWITCHES -

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

2.4.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>) - <UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIALOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

2.4.2.3 EFFECT OF RESTART COMMAND -

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

2.4.3 CONTINUE COMMAND

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

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

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

2.4.3.2 EFFECT OF CONTINUE COMMAND -

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

2.4.4 PROCEED COMMAND -

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

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

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

2.4.4.2 EFFECT OF PROCEED COMMAND -

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

2.4.5 ADD COMMAND -

ADD/UNITS:<UNIT-LIST>

2.4.6 EFFECT OF ADD COMMAND -

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

2.4.7 DROP COMMAND

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

2.4.8 EFFECT OF DROP COMMAND -

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

2.4.9 PRINT COMMAND -

PRI(NT)

2.4.9.1 EFFECT OF PRINT COMMAND -

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

2.4.10 DISPLAY COMMAND -

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

2.4.10.1 EFFECT OF DISPLAY COMMAND -

THE HARDWARE P TABLE FOR THE TEST STATION IS PRINTED IN THE
FORMAT IN WHICH IT WAS ENTERED.

2.4.11 FLAGS COMMAND -

FLA(GS)

2.4.11.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

2.4.12 ZFLAGS COMMAND

ZFL(AGS)

2.4.13 ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

2.4.14 CONTROL CHARACTERS -

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

2.5 HARDWARE QUESTIONS

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

1. CSR ADDRESS - THIS QUESTION REQUESTS THE CSR ADDRESS OF THE SPECIFIED 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. TYPE OF LOOPBACK (1=INTERNAL, 2=H3029 OR H3277) - THIS QUESTION REQUESTS THE TYPE OF LOOPBACK TO BE USED WHEN TESTING THE DHU-11. THE FOLLOWING TYPES ARE SUPPORTED:
 - 0 INTERNAL - ONLY INTERNAL UART LOOPBACK IS TO BE USED IN TESTING THE DHU-11.
 - 0 H3029 OR H3277 - STAGGERED LOOPBACK CONNECTORS ARE PROVIDED ON THE DHU11 DISTRIBUTION PANEL (H3029) IF THIS DISTRIBUTION PANEL IS NOT PRESENT THEN H3277 STAGGERED BERG CONNECTOR(S) MUST BE INSTALLED ON THE BERG CONNECTOR SOCKETS OF THE DHU11.
 - 0 H325 - SINGLE LINE, 25 PIN LOOPBACK CONNECTORS (TYPE H325) ARE INSTALLED ON THE LINES TO BE TESTED.

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. 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.
3. 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.

Q UNITS (D) ? 8<CR>

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

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

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

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

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

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

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

```
UNIT 8
CSR ADDRESS (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:

CZDHV DVC FTL ERR 04106 ON UNIT 00 TST 003 SUB 000 PC: XXXXXX
DMA_ABORT BIT TEST FAILED

THIS ERROR INDICATES THAT A FATAL ERROR WAS ENCOUNTERED DURING THE TEST WHICH TESTS THE DMA_ABORT BIT.

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

CZDHV DVC FTL ERR 04106 ON UNIT 00 TST 003 SUB 000 PC: XXXXXX
DMA_ABORT BIT TEST FAILED
DMA_START BIT FOUND SET AFTER DMA ABORTED ON LINE: 8

4.0 PERFORMANCE AND PROGRESS REPORTS

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

5.0 TEST SUMMARIES

THE FOLLOWING ARE INCLUDED WITHIN CZDHVB:

1. DEVICE REGISTER ACCESS TEST - VERIFIES THAT THE UUT REGISTERS WILL RESPOND WITH THE CORRECT UNIBUS HANDSHAKING SIGNALS. VERIFIES THAT THE UUT IS AT THE CORRECT ADDRESS.
2. DMA START TEST - VERIFIES THAT EACH DMA START BIT WILL INITIATE A DMA TRANSMISSION ON A LINE
3. DMA ABORT TEST - VERIFIES THAT EACH DMA ABORT BIT WILL STOP A DMA TRANSMISSION, RETURN A TX ACTION AND SUCCESSFULLY RESTART THE DMA.
4. DMA ERROR TEST - VERIFIES THAT THE DMA ERROR BIT IN THE CSR REPORTS DMA ERRORS CORRECTLY WHEN THEY OCCUR.
5. O.AUTO INACTIVE TEST - VERIFIES THAT THE DUT WILL NOT RESPOND TO INCOMING XON AND XOFF CHARACTERS WHEN O.AUTO IS CLEAR.
6. O.AUTO ACTIVE TEST - VERIFIES THAT THE DUT RESPONDS CORRECTLY TO INCOMING FLOW CONTROL CHARACTERS WHEN ACTIVE
7. I.AUTO INACTIVE TEST - VERIFIES THAT THE DUT WILL NOT GENERATE XON AND XOFF CHARACTERS IN RESPONSE TO THE APPROPRIATE FIFO CONDITIONS WHEN I.AUTO IS INACTIVE.
8. I.AUTO ACTIVE TEST - VERIFIES THAT THE DUT WILL GENERATE XON AND XOFF CHARACTERS IN RESPONSE TO THE APPROPRIATE FIFO CONDITIONS WHEN I.AUTO IS ACTIVE.
9. FIFO DATA TEST - VERIFIES THAT THE FIFO WILL HOLD 256 CHARACTERS WITHOUT CORRUPTING DATA.
10. FIFO 3/4 LEVEL INACTIVE TEST - VERIFIES THAT THE 3/4 LEVEL ALARM DOES NOT BECOME ACTIVE BELOW THE 3/4 LEVEL.
11. FIFO 3/4 LEVEL ACTIVE TEST - VERIFIES THAT THE 3/4 LEVEL ALARM BECOMES ACTIVE WHEN THE FIFO IS 3/4 FULL.
12. FIFO 3/4 LEVEL ACTIVE/INACTIVE TEST - VERIFIES THAT THE 3/4 LEVEL ALARM, ONCE ACTIVATED, REMAINS ACTIVE UNTIL THE FIFO IS REDUCED BELOW THE 1/2 LEVEL.
13. FIFO 1/2 LEVEL TEST - VERIFIES THAT THE FIFO 1/2 LEVEL ALARM SYSTEM BECOMES ACTIVE AND INACTIVE AT THE CORRECT LEVELS.
14. RXTIMER TEST - VERIFIES THAT THE HOLD OFF TIMER FOR RX INTERRUPTS IS OPERATING CORRECTLY, AND THAT THE 3/4 FULL LEVEL OVERRIDES THE TIMER.

15. TX-ACTION FIFO TEST - VERIFIES THAT THE TX-ACTION FIFO CAN HOLD 16 UNIQUE TX-ACTIONS, AND THAT ONLY ONE TX INTERRUPT OCCURS FOR ALL 16 TX-ACTIONS.
16. TX FIFO TEST - VERIFIES THAT THE FIFO WILL 64 UNIQUE CHARACTERS AND ALSO THAT ONLY ONE INTERRUPT OCCURS FOR ALL 64 CHARACTERS.
17. BREAK GENERATION TEST - VERIFIES THAT ALL SERIAL TRANSMIT LINES CAN GENERATE A BREAK BY SETTING THE BRK BIT IN THE ASSOCIATED LNCTRL REGISTER.
18. NO OVERRUN ERROR TEST - VERIFIES THAT THE DUT WILL NOT REPORT DATA OVERRUN ERRORS WHEN THEY DO NOT OCCUR.
19. OVERRUN ERROR TEST - VERIFIES THAT THE DUT WILL REPORT DATA OVERRUN ERRORS WHEN THEY OCCUR.
20. REPORT BMP CODES TEST - THIS PSEUDO TEST REPORTS THE FIRST 32 CHARACTERS WHICH WERE DISCOVERED IN THE FIFO DURING THE EXECUTION OF THE OTHER TESTS. THIS AVOIDS INTERRUPTION OF THE OTHER TESTS BY THESE CODES IF THEY ARE NOT CRITICAL TO THE PERFORMANCE OF THE TESTS.

6.0 EXAMPLE ERROR FREE PASS

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

```
DR CZDHVBO
CZDHVBO.BIN
DRS
CZDHV-B-0
DHU-11 FUNC TST PART2
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>
TYPE OF LOOPBACK (1=INTERNAL, 2=H3029 OR H3277): (0) 2 ? 1
```

CHANGE SW (L) ? Y

```
REPORT UNIT NUMBER AS EACH UNIT IS TESTED: (L) Y ? <CR>
EXTENDED ERROR REPORTING: (L) N ? <CR>
```

TESTING UNIT : 0

TESTING UNIT : 1

```
CZDHV EOP      1
0 TOTAL ERRS
```

DR>

```

1050          .LIST SEQ,LOC,BIN,MEB
1051          .NLIST CND
1059
1060
1061          .SBTTL  PROGRAM HEADER
1062
1063
1064          .MCALL  SVC
1065 000000          SVC                      ; INITIALIZE SUPERVISOR MACROS
1066
1067          ;*****
1068          ; IF STRUCTURED MACROS ARE TO BE USED, ADD ".MCALL STRUCT" AND "STRUCT"
1069          ; TO INITIALIZE THE STRUCTURED MACROS.
1070
1071          SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
1072          SVCTST= 1      ; LIST TEST TAGS, SHIFTED RIGHT
1073          SVCSUB= 1      ; LIST SUBTEST TAGS, SHIFTED RIGHT
1074          SVCGBL= 1      ; LIST GLOBAL TAGS, SHIFTED RIGHT
1075          SVCTAG= 1      ; LIST OTHER TAGS, SHIFTED RIGHT
1076
1077          ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1078          ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
1079          ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
1080          ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1081          ;*****
1082
1083 000000          .ENABL  ABS
1084          ;.ENABL  AMA
1085          "      "      2000
1086
1087 002000          BGNMOD
1088
1089          ;**
1090          ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1091          ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1092          ;--
1093
1094 002000          POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL
1095
1112
1113 002000          HEADER  CZDHY,B,0,22,0,PRI07
1114
1115          L$NAME::
1116          .ASCII  /C/
1117          .ASCII  /Z/
1118          .ASCII  /D/
1119          .ASCII  /H/
1120          .ASCII  /V/
1121          .BYTE   0
1122          .BYTE   0
1123          .BYTE   0
1124
1125          L$REV::
1126          .ASCII  /B/
1127
1128          L$DEPO::
1129          .ASCII  /O/
1130
1131          L$UNIT::
1132          .WORD   0
1133
1134          L$TIML::

```

```

002014 000022
002016
002016 035156
002020
002020 035404
002022
002022 002176
002024
002024 002210
002026
002026 035706
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 000000
002056
002056 000000
002060
002060 004120
002062
002062 020030
002064
002064 000000
002066
002066 000000
002070
002070 000000
002072
002072 020704
002074
002074 000000
002076
002076 004130
002100
002100 104035
002102
002102 004050
002104
002104 020044

```

```

L$HPCP:: .WORD 22
L$HPCP:: .WORD L$HARD
L$SPCP:: .WORD L$SOFT
L$HPTP:: .WORD L$HW
L$SPTP:: .WORD L$SW
L$LADP:: .WORD L$LAST
L$STA:: .WORD 0
L$CO:: .WORD 0
L$DTYP:: .WORD 0
L$APT:: .WORD 0
L$DTP:: .WORD L$DISPATCH
L$PRIO:: .WORD PRI07
L$ENVI:: .WORD 0
L$EXP1:: .WORD 0
L$MREV:: .WORD 0
L$EF:: .BYTE C$REVISION
        .BYTE C$EDIT
L$EF:: .WORD 0
        .WORD 0
L$SPC:: .WORD 0
L$DEVP:: .WORD L$DVTYP
L$REPP:: .WORD L$RPT
L$EXP4:: .WORD 0
L$EXP5:: .WORD 0
L$AUT:: .WORD 0
L$DUT:: .WORD L$DU
L$LUN:: .WORD 0
L$DESP:: .WORD L$DESC
L$LOAD:: EMT E$LOAD
L$ETP:: .WORD L$ERRTBL
L$ICP:: .WORD L$INIT

```

N2

CZDHV80 DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 23-2

SEQ 26

002106
002106 020666
002110
002110 020664
002112
02112 020036
002114
002114 000000
002116
002116 000000
002120
002120 000000

1114

L\$CCP::
L\$ACP:: .WORD L\$CLEAN
L\$PRT:: .WORD L\$AUTO
L\$TEST:: .WORD L\$PROT
L\$DLY:: .WORD 0
L\$HIME:: .WORD 0

B3

CZDHVBO DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 24
DISPATCH TABLE

SEQ 27

1126
1127
1128
1129
1130
1131
1132
1133

.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 20

002122
002122 000024
002124
002124 021022
002126 021304
002130 021710
002132 022346
002134 022674
002136 023474
002140 024274
002142 024716
002144 025410
002146 025716
002150 026226
002152 026726
002154 027424
002156 030102
002160 031152
002162 031740
002164 032766
002166 033412
002170 034130
002172 035074
1134

.WORD 20
L1DISPATCH::
.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

C3

670HV80.DHL.11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 25

SEQ 28

```

1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160 002174
      002174 000004
      002176
      002176
1161
1162 002176 160460
1163 002200 000310
1164 002202 177777
1165 002204 002
1166
1167 002206
      002206

      .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.
      ;**
      BGNHW  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  2      ;DEFAULT LOOPBACK MODE
      .EVEN
      ENDHW
                                     L10000:

```

D3

GZPHVBO DMU 11 FUNC 1ST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 26

SEQ 29

```

1176
1177
1178          .SBTTL  SOFTWARE P-TABLE
1179
1180          ;**
1181          ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
1182          ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
1183          ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
1184          ; AT RUN TIME.
1185          ;--
1186
1187          BGNSW  SFPTBL
1188
1189          002206 000002
1190          002210
1191          002210
1192          002214
1193          002214
1194
1195          OPTION:: .WORD 20 ;BIT MAP OF PROGRAM CONTROL FLAGS
1196          NDERPT:: .WORD 0  ;DEFAULT NUMBER OF INDIVIDUAL DATA ERRORS TO RPT.
1197
1198          ENDSW
1199
1200          L10001:
1201
1202          L10001: .WORD L10001-L10001/2
1203          SFPTBL::
1204
1205          L10001:

```

```

1201
1202
1203
1213
1214
1215
1216
1217
1218
1219
1220
1221      000020
1222      177777
1223
1224
1225      000000
1226      000002
1227      000002
1228      000004
1229      000006
1230      000006
1231      000010
1232      000012
1233      000014
1234      000016
1235
1236
1237      000020
1238      000030
1239      000100
1240
1241
1256 002214

      .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 8.
      MAPLNS==177777  ;BIT MAP OF LINES ON DHU11.

      ;***** DEVICE REGISTER OFFSETS FROM THE CSR'S ADDRESS *****
      CSRO==0          ;CSR REGISTER OFFSET FROM THE CSR ADDRESS
      RBUFO==2         ;RECEIVE REGISTER OFFSET FROM THE CSR ADDRESS
      RXTIMO==2        ;RECIEVE TIMER REGISTER OFFSET FROM THE CSR ADDRESS
      LPRO==4          ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
      FLSO==6          ;FIFO SIZE/STATUS REGISTER OFFSET FROM THE CSR ADDRESS
      FDATA==6         ;FIFO DATA REGISTER OFFSET FROM THE CSR ADDRESS
      LNCTRO==10       ;LINE CONTROL REGISTER OFFSET FROM THE CSR ADDRESS
      TXAD10==12        ;TRANSMIT ADDRESS 1 REGISTER OFFSET FROM THE CSR ADDRESS
      TXAD20==14        ;TRANSMIT ADDRESS 2 REGISTER OFFSET FROM THE CSR ADDRESS
      TXBFCO==16       ;TRANSMIT COUNT REGISTER OFFSET FROM THE CSR ADDRESS

      ;***** EQUATES USED WITH RESPECT TO THE RX BUFFER *****
      RXBETX==16.      ;LEVEL OF RX BUFFER AT WHICH TO RE-ENABLE TRANSMISSION.
      RXBDTX==24.      ;LEVEL OF RX BUFFER AT WHICH TO DISABLE TRANSMISSION.
      RXBFUL==64.      ;TOTAL CHARACTER CAPACITY OF THE RX BUFFER.

      EQUALS

      ;
      ; BIT DIFINITIONS
      ;
      BIT15== 100000
      BIT14== 40000
      BIT13== 20000
      BIT12== 10000
      BIT11== 4000
      BIT10== 2000
      BIT09== 1000
      BIT08== 400
      BIT07== 200
      BIT06== 100
      BIT05== 40
      BIT04== 20
      BIT03== 10
      BIT02== 4
      BIT01== 2
      BIT00== 1

      ;
      BIT9== BIT09
      BIT8== BIT08
      BIT7== BIT07
      BIT6== BIT06

```

F3

CZDHVRO DHU-11 FUNC.1ST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 27-1

SEQ 31

GLOBAL EQUATES SECTION

```

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

;
; EVENT FLAG DEFINITIONS
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
;
000040      EF.START== 32.      ; START COMMAND WAS ISSUED
000037      EF.RESTART== 31.    ; RESTART COMMAND WAS ISSUED
000036      EF.CONTINUE== 30.   ; CONTINUE COMMAND WAS ISSUED
000035      EF.NEW== 29.        ; A NEW PASS HAS BEEN STARTED
000034      EF.PWR== 28.        ; A POWER-FAIL/POWER-UP OCCURRED
;
;
; PRIORITY LEVEL DEFINITIONS
;
000340      PRI07== 340
000300      PRI06== 300
000240      PRI05== 240
000200      PRI04== 200
000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0

;
; OPERATOR FLAG BITS
;
000004      EVL== 4
000010      LOT== 10
000020      ADR== 20
000040      IDU== 40
000100      ISR== 100
000200      UAM== 200
000400      BOE== 400
001000      PNT== 1000
002000      PRI== 2000
004000      IXE== 4000
010000      IBE== 10000
020000      IER== 20000
040000      LOE== 40000
100000      HOE== 100000

```

1257

1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279 002214 000200
1280 002216 000204
1281 002220 177777
1282 002222 000
1283 002223 004
1284 002224 000000
1285
1286
1287
1288
1289 002226
1290 002226 160020
1291 002230 160022
1292 002232 160024
1293 002234 160026
1294
1295 002236 160030
1296 002240 160032
1297 002242 160034
1298 002244 160036
1299
1300
1301
1302
1303 002246 000000
1304 002250 000000
1305 002252 000000
1306 002254 000001
1307 002256 000000
1308 002260 031463
1309 002262 146314
1310 002264 000000
1311 002266 000000
1312 002270 000000
1313 002272 000000
1314 002274 000000
1315 002276 000000
1316 002300 000000
1317 002302 000000
1318
1319
1320
1321 002304 177546
1322 002306 000300

.SBTTL GLOBAL DATA SECTION

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

; UNIT VARIABLE AREA

RXVECA:: .WORD 200 ;RX VECTOR ADDRESS.
TXVECA:: .WORD 204 ;TX VECTOR ADDRESS.
ACTLNS:: .WORD 177777 ;ACTIVE LINE BIT MAP.
LOPCK:: .BYTE 0 ;LOOPBACK MODE
BRLEV:: .BYTE 4 ;INTERRUPT BUS REQUEST LEVEL
UNITN:: .WORD 0 ;UNIT NUMBER.

; 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

; ASSORTED GLOBAL VARIABLES:

BUFPTR:: .WORD 0 ;STORAGE FOR RECEIVE CHARACTER BUFFER POINTER.
CTRLCF:: .WORD 0 ;STORAGE FOR THE CONTROL-C FLAG.
EXERR:: .WORD 0 ; "EXIT ON ERROR" FLAG.
TSTNUM:: .WORD 1 ;STORAGE FOR THE TEST NUMBER.
IESTAT:: .WORD 0 ;STORAGE FOR STATES OF THE DUT INT ENABLE BITS.
LGRP1M:: .WORD 31463 ;BIT MAP OF LINES IN LINE GROUP I.
LGRP2M:: .WORD 146314 ;BIT MAP OF LINES IN LINE GROUP II.
PASCNT:: .WORD 0 ;STO'G FOR PASS COUNT USED IN ROM VERSION# TST.
RXINTC:: .WORD 0 ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
RXINTF:: .WORD 0 ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
TXINTC:: .WORD 0 ;STORAGE FOR TRANSMIT INTERRUPT COUNT.
TXINTF:: .WORD 0 ;STORAGE FOR TRANSMIT INTERRUPT FLAGS.
IP4VEC:: .WORD 0 ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
IP4FLG:: .WORD 0 ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
WORD1:: .WORD 0 ;LOCATION FOR PASSING INDIRECT PARAMETERS.

; LINE TIME CLOCK VARIABLES AND STORAGE.

CLKCSR:: .WORD 177546 ;CSR ADDRESS OF THE LTC.
CLKBRL:: .WORD PRI06 ;INTERRUPT PRIORITY LEVEL OF THE LTC.

```

1323 002310 000100          CLKVEC:: .WORD 100      ;INTERRUPT VECTOR ADDRESS OF THE LTC.
1324 002312 000074          CLKHRZ:: .WORD 60.       ;INTERRUPT FREQUENCY OF THE LTC.
1325 002314 000000          TIMER1:: .WORD 0         ;HARDWARE CLOCK COUNTER #1.
1326 002316 000000          TIMER2:: .WORD 0         ;HARDWARE CLOCK COUNTER #2.
1327 002320 000170          TIMER3:: .WORD 120.       ;HARDWARE BREAK COUNTER LOCATION.
1328 002322 000170          BCOUNT:: .WORD 120.       ;BREAK COUNT VALUE IN CLOCK TICKS.
1329 002324 000021          MSTICK:: .WORD 17.        ;NUMBER OF MILLI-SECONDS PER LTC TICK.
1330 002326 000062          MSLCNT:: .WORD 62         ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.
1331
1332
1333          ;*****
1334          ;      MEMORY MANAGEMENT VARIABLES AND FLAGS.
1335          ;*****
1335 002330 177572          MMSRO:: .WORD 177572      ;ADDRESS OF MEM MGT STATUS REGISTER #0.
1336 002332 000000          MMPRES:: .WORD 0         ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
1337 002334 000000          MMENAB:: .WORD 0         ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).
1338
1339          ;*****
1340          ;      TABLE OF WORDS WITH CORRESPONDING BIT SET FOR GENERATION OF BIT MAPS.
1341          ;*****
1342 002336 000001          BITTBL:: .WORD 1           ;BIT 0 SET.
1343 002340 000002          .WORD 2           ;BIT 1 SET.
1344 002342 000004          .WORD 4           ;BIT 2 SET.
1345 002344 000010          .WORD 10          ;BIT 3 SET.
1346 002346 000020          .WORD 20          ;BIT 4 SET.
1347 002350 000040          .WORD 40          ;BIT 5 SET.
1348 002352 000100          .WORD 100         ;BIT 6 SET.
1349 002354 000200          .WORD 200         ;BIT 7 SET.
1350 002356 000400          .WORD 400         ;BIT 8 SET.
1351 002360 001000          .WORD 1000        ;BIT 9 SET.
1352 002362 002000          .WORD 2000        ;BIT 10 SET.
1353 002364 004000          .WORD 4000        ;BIT 11 SET.
1354 002366 010000          .WORD 10000       ;BIT 12 SET.
1355 002370 020000          .WORD 20000       ;BIT 13 SET.
1356 002372 040000          .WORD 40000       ;BIT 14 SET.
1357 002374 100000          .WORD 100000      ;BIT 15 SET.
1358
1359          ;*****
1360          ;      GPR SAVE AREAS ZERO AND ONE.
1361          ;*****
1362 002376          GPRSOB::          ;BASE OF GPR SAVE AREA NUMBER ZERO.
1363 002376 000000          .WORD 0           ;WORD 1, STORAGE FOR R1.
1364 002400 000000          .WORD 0           ;WORD 2, STORAGE FOR R2.
1365 002402 000000          .WORD 0           ;WORD 3, STORAGE FOR R3.
1366 002404 000000          .WORD 0           ;WORD 4, STORAGE FOR R4.
1367 002406 000000          .WORD 0           ;WORD 5, STORAGE FOR R5.
1368
1369          ;*****
1370          ;      STORAGE AREA FOR THE BMP CODE QUEUE.
1371          ;*****
1371 002410 000000          BMPCQP:: .WORD 0         ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
1372 002412          BMPCQB:: .BLKW 64.          ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
1373 002612          BMPCQE::          ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1374
1375          ;*****
1376          ;      STORAGE AREA FOR ERROR SUMMARY TABLE AND FLAGS.
1377          ;*****
1378 002612 000000          ERSMRF:: .WORD 0         ;ERROR SUMMARY FLAGS.
1379 002614          ERCNTB:: .BLKW 16          ;TABLE OF ERROR COUNTS.

```

```

1380
1381
1382 ;*****
1383 ; GENERAL TABLE AND BUFFER AREA--513 WORDS.
1384 ;*****
1384 002650 BUFBAS:: ;BASE OF MEMORY BUFFER.
1385 002650 ERLTBL:: .BLKW 128. ;FIRST HALF OF GENERAL TABLE OR BUFFER.
1386 003250 BUFMID:: .BLKW 64. ;SECOND HALF OF GENERAL TABLE OR BUFFER.
1387 003450 BUF3QT:: .BLKW 64. ;LAST QUARTER OF THE BUFFER AREA.
1388 003650 BUFEND:: ;END OF GENERAL PURPOSE MEMORY BUFFER.
1389 003650 ENDETBL:: .BLKW 16. ;BUFFER OVERFLOW SPACE.
1390
1391 ;*****
1392 ; RECEPTION TABLE OF COUNTERS
1393 ;*****
1394 003710 RXCNTB:: .BLKW 16. ;RECEPTION CHARACTER COUNTERS TABLE.
1395
1396 ;*****
1397 ; * TABLE FOR STORAGE OF RX/TX LINE NUMBER ASSOCIATIONS.
1398 ; * THE ASSOCIATIONS ARE STORED AS LINE NUMBER TIMES 2 FOR USE AS OFFSETS
1399 ; * WHEN ACCESSING A TABLE OF WORDS.
1400 ; * NOTE: DO NOT WRITE A NON-ZERO VALUE INTO THE UPPER BYTE OF ANY ENTRY.
1401 ;*****
1402 003750 TXRXLB:: ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1403 003750 000000 .WORD 0 ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
1404 003752 000002 .WORD 2. ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
1405 003754 000004 .WORD 4. ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
1406 003756 000006 .WORD 6. ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
1407 003760 000010 .WORD 8. ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
1408 003762 000012 .WORD 10. ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
1409 003764 000014 .WORD 12. ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
1410 003766 000016 .WORD 14. ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
1411 003770 000020 .WORD 16. ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
1412 003772 000022 .WORD 18. ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
1413 003774 000024 .WORD 20. ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
1414 003776 000026 .WORD 22. ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
1415 004000 000030 .WORD 24. ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
1416 004002 000032 .WORD 26. ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
1417 004004 000034 .WORD 28. ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
1418 004006 000036 .WORD 30. ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
1419 004010 TXRXLE:: ;END OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1420 .EVEN ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1421 ;*****
1422 ; * TABLE FOR STORAGE OF RX/TX LINE NUMBER ASSOCIATIONS.
1423 ; * THE ASSOCIATIONS ARE STORED AS LINE NUMBERS WHICH CAN BE USED AS SUCH OR
1424 ; * AS OFFSETS WHEN ACCESSING A TABLE OF BYTES.
1425 ;*****
1426 004010 TXRLNB:: ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1427 004010 000 .BYTE 0 ;TX/RX LINE FOR RX/TX LINE 0.
1428 004011 001 .BYTE 1. ;TX/RX LINE FOR RX/TX LINE 1.
1429 004012 002 .BYTE 2. ;TX/RX LINE FOR RX/TX LINE 2.
1430 004013 003 .BYTE 3. ;TX/RX LINE FOR RX/TX LINE 3.
1431 004014 004 .BYTE 4. ;TX/RX LINE FOR RX/TX LINE 4.
1432 004015 005 .BYTE 5. ;TX/RX LINE FOR RX/TX LINE 5.
1433 004016 006 .BYTE 6. ;TX/RX LINE FOR RX/TX LINE 6.
1434 004017 007 .BYTE 7. ;TX/RX LINE FOR RX/TX LINE 7.
1435 004020 010 .BYTE 8. ;TX/RX LINE FOR RX/TX LINE 8.
1436 004021 011 .BYTE 9. ;TX/RX LINE FOR RX/TX LINE 9.

```

```

1437 004022      012      .BYTE 10.      ;TX/RX LINE FOR RX/TX LINE 10.
1438 004023      013      .BYTE 11.      ;TX/RX LINE FOR RX/TX LINE 11.
1439 004024      014      .BYTE 12.      ;TX/RX LINE FOR RX/TX LINE 12.
1440 004025      015      .BYTE 13.      ;TX/RX LINE FOR RX/TX LINE 13.
1441 004026      016      .BYTE 14.      ;TX/RX LINE FOR RX/TX LINE 14.
1442 004027      017      .BYTE 15.      ;TX/RX LINE FOR RX/TX LINE 15.
1443 004030      .EVEN      ;END OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1444      .EVEN      ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1445      ;*****
1446      ;* TABLE OF TX/RX LINE NUMBER ASSOCIATIONS IN STAGGERED LOOPBACK.
1447      ;* THE ASSOCIATIONS ARE STORED AS LINE NUMBER TIMES 2 FOR USE AS OFFSETS
1448      ;* WHEN ACCESSING A TABLE OF WORDS.
1449      ;* THIS IS A TABLE OF DATA FOR READING ONLY. USE TO LOAD THE ABOVE TABLE.
1450      ;* NOTE: MUST CONVERT FROM BYTES TO WORDS WHEN LOADING ABOVE TABLE.
1451      ;*****
1452 004030      STGTRB:;      ;BASE OF STAGGERED TX/RX LINE NUMBER TABLE.
1453 004030      004      .BYTE 4.      ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
1454 004031      006      .BYTE 6.      ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
1455 004032      000      .BYTE 0.      ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
1456 004033      002      .BYTE 2.      ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
1457 004034      014      .BYTE 12.     ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
1458 004035      016      .BYTE 14.     ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
1459 004036      010      .BYTE 8.      ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
1460 004037      012      .BYTE 10.     ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
1461 004040      024      .BYTE 20.     ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
1462 004041      026      .BYTE 22.     ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
1463 004042      020      .BYTE 16.     ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
1464 004043      022      .BYTE 18.     ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
1465 004044      034      .BYTE 28.     ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
1466 004045      036      .BYTE 30.     ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
1467 004046      030      .BYTE 24.     ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
1468 004047      032      .BYTE 26.     ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
1469      .EVEN      ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1482 004050      ERTTBL      ;
      004050      L$ERTTBL:;
      004050      000000      ERRTP:;      .WORD 0
      004052      000000      ERRNBR:;      .WORD 0
      004054      000000      ERRMSG:;      .WORD 0
      004056      000000      ERRBLK:;      .WORD 0
1483      .EVEN
1484

```

```

1486      ,SBTTL  GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.
1487      ;*****
1488      ;*      THERE ARE 4 ROUTINES AND MACRO DEFINITIONS USED FOR THE HANDLING OF
1489      ;*      GPR VALUES DURING SUBROUTINE CALLS WITHIN THIS PROGRAM.  THE FOUR
1490      ;*      ROUTINES/MACRO CALLS HAVE THE FOLLOWING NAMES:
1491      ;*
1492      ;*      SAVE - MACRO DEFINITION USED AT THE BEGINNING OF A SUBROUTINE TO
1493      ;*      SAVE THE GPR CONTENTS FOR LATER RESTORATION.
1494      ;*      PASS - MACRO DEFINITION USED AT THE END OF A SUBROUTINE TO RESTORE
1495      ;*      THE PREVIOUSLY SAVED GPR CONTENTS AND TO LEAVE THE CONTENTS
1496      ;*      OF THE SPECIFIED GPR(S) INTACT (NOT RESTORED).
1497      ;*      PREG05 - SUBROUTINE WHICH IS CALLED FROM THE SAVE AND PASS MACRO
1498      ;*      EXPANSIONS WHICH ACTUALLY PERFORMS THE ACTIONS ON THE GPRS.
1499      ;*
1500      ;*      DURING A SUBROUTINE WHICH USES THESE GPR SAVE ROUTINES THE VALUES
1501      ;*      OF THE GPRS ARE STORED ON THE STACK IN THE FOLLOWING STACK FRAME:
1502      ;*
1503      ;*      SP      -> RET PC INTO PREG05 ROUTINE.
1504      ;*      SP+2    -> GPR R0 CONTENTS.
1505      ;*      SP+4    -> GPR R1 CONTENTS.
1506      ;*      SP+6    -> GPR R2 CONTENTS.
1507      ;*      SP+8    -> GPR R3 CONTENTS.
1508      ;*      SP+10   -> GPR R4 CONTENTS.
1509      ;*      SP+12   -> GPR R5 CONTENTS.
1510      ;*      SP+14   -> RET PC INTO CALLER OF SUB'TNE WHICH CALLED PREG05.
1511      ;*
1512      ;*      EACH LEVEL OF SUB'TNE CALLING USES 8 WORDS OF STACK OVERHEAD.
1513      ;*      THE SAVE AND PASS MACROS CAN ALSO BE USED IN "STRAIGHT LINE CODE"
1514      ;*      TO SAVE AND RESTORE THE GPR VALUES.  IN ANY CASE, AFTER THE
1515      ;*      ISSUING OF A PASS CALL THE GPRS WILL BE RESTORED TO THE VALUES
1516      ;*      THEY HAD PRIOR TO THE LAST SAVE CALL (EXCEPT FOR THE EXCEPTED,
1517      ;*      OR PASSED INTACT, GPRS SPECIFIED AS PARAMETERS TO THE PASS CALL)
1518      ;*      AND THE SP WILL ALSO BE RESTORED TO ITS CONDITION BEFORE THE LAST
1519      ;*      SAVE CALL.  THE PROGRAMMER MUST BE SURE THAT THE SP HAS THE SAME
1520      ;*      VALUE WHEN THE PASS MACRO IS CALLED AS IT HAD IMMEDIATELY AFTER
1521      ;*      THE SAVE MACRO WAS CALLED.
1522      ;*****

```

L3

```
1524 .SBITL GPR FRAME ACCESS EQUATES
1525      ;***
1526      ;EQUATES THAT ALLOW ACCESS TO THE STACK FRAME. THESE ARE THE
1527      ;OFFSETS INTO THE STACK FOR REGISTERS SAVED DURING THE PREG05
1528      ;ROUTINE.
1529      ;***
1530
1531      000036      LPCSLT**      36      ;OFFSET FOR LAST RETURN PC.
1532      000016      PCSLOT**      16      ;OFFSET FOR RETURN PC.
1533      000014      R5SLOT**      14      ;OFFSET FOR R5.
1534      000012      R4SLOT**      12      ;OFFSET FOR R4.
1535      000010      R3SLOT**      10      ;OFFSET FOR R3.
1536      000006      R2SLOT**      6       ;OFFSET FOR R2.
1537      000004      R1SLOT**      4       ;OFFSET FOR R1.
1538      000002      R0SLOT**      2       ;OFFSET FOR R0.
```

M3

CZDHV80 DHU-11 FUNC 1ST PART2 MACRO M120C 15-MAR-84 09:28 PAGE 31
GLOBAL MACRO DEFINITION - SAVE -

SEQ 38

```

1540 .SBTTL GLOBAL MACRO DEFINITION - SAVE -
1541 ;*****
1542 ;* THIS MACRO IS USED AT THE BEGINNING OF A SUBROUTINE TO SAVE THE
1543 ;* CONTENTS OF THE GPRS R0 THRU R5.
1544 ;*
1545 ;* INPUTS: SP - UNCHANGED SINCE SUBROUTINE WAS ENTERED
1546 ;* R5SLOT - OFFSET TO STACK SLOT FOR R5 (EQUATED TO 14 OCTAL)
1547 ;*
1548 ;* OUTPUTS: GPR SAVE AREA ON THE STACK IS LOADED WITH THE CONTENTS OF GPRS
1549 ;* TOP OF STACK - LOADED WITH THE RETURN ADDRESS INTO PREG05
1550 ;*
1551 ;* CALLING SEQUENCE: SAVE
1552 ;*
1553 ;* COMMENTS: NO ARGUMENTS ARE ALLOWED.
1554 ;* THE PASS MACRO SHOULD BE CALLED TO RESTORE THE GPR VALUES.
1555 ;*
1556 ;* SUBORDINATE ROUTINES CALLED: PREG05.
1557 ;*****
1558 .MACRO SAVE
1559 .LIST
1560 JSR R5,PREG05 ;CALL REGISTER SAVE SUBPT.
1561 .NLIST
1562 .ENDM SAVE
1563

```

```

1565 .SBTTL GLOBAL MACRO DEFINITION - PASS -
1566 ;*****
1567 ;* THIS MACRO IS USED IN CONJUNCTION WITH THE SAVE MACRO. IT IS
1568 ;* CALLED AT END OF A SUBROUTINE TO PASS PARAMETERS IN GPRS BACK TO THE
1569 ;* CALLING ROUTINE BY ALTERING THE GPR SAVE AREA ON THE STACK AND THEN
1570 ;* RETURNING TO PREG05 TO RESTORE THE GPRS TO THEIR SAVED VALUES.
1571 ;*
1572 ;* INPUTS: ONLY ALLOWED ARGUMENTS ARE "R0" THRU "R5".
1573 ;* ROSLOT THRU R5SLOT MUST BE EQUATED TO THEIR RESPECTIVE GPR SAVE
1574 ;* SLOT OFFSETS BEFORE CALLING THIS MACRO.
1575 ;*
1576 ;* OUTPUTS: THE GPR VALUES ARE PUT IN THEIR RESPECTIVE SLOTS ON THE STACK.
1577 ;*
1578 ;* CALLING SEQUENCE: PASS R0,R1,...
1579 ;*
1580 ;* COMMENTS: ANY COMBINATION OF GPR ARGUMENTS MAY BE LISTED IN ANY ORDER.
1581 ;* FOR EXAMPLE, THE FOLLOWING ARE LEGAL:
1582 ;* PASS R1
1583 ;* PASS R4,R0,R2
1584 ;* THE GPRS LISTED AS ARGUMENTS WILL BE PASSED INTACT TO THE
1585 ;* CALLING ROUTINE, ALL OTHER GPRS WILL BE RESTORED.
1586 ;* THE SP MUST BE AT ITS ORIGINAL VALUE WHEN PASS IS CALLED.
1587 ;*
1588 ;* THE MACRO CALL
1589 ;* PASS R0,R3
1590 ;* EXPANDS INTO THE FOLLOWING ASSEMBLY CODE:
1591 ;* MOV R0,ROSLOT(SP) ;PUT R0 IN STACK SLOT.
1592 ;* MOV R3,R3SLOT(SP) ;PUT R3 IN STACK SLOT.
1593 ;* JSR PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
1594 ;* IN THIS EXAMPLE GPRS R1, R2, R4, AND R5 WILL BE RESTORED TO
1595 ;* THEIR VALUES CONTAINED IN THE STACK FRAME AND R0 AND R3
1596 ;* WILL BE LEFT AT THEIR VALUES PRIOR TO THIS PASS CALL
1597 ;*
1598 ;* SUBORDINATE ROUTINES CALLED: (PREGRT - LABEL WITHIN PREG05, VALUE ON STACK.)
1599 ;*****
1600
1601 .MACRO PASS A,B,C,D,E,F
1602 .IRP X,<A,B,C,D,E,F>
1603 .IF NB,X
1604 .LIST
1605 MOV X,X'SLOT(SP) ;PUT X IN STACK SLOT.
1606 .NLIST
1607 .ENDC
1608 .ENDM
1609 .LIST
1610 JSR PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
1611 .NLIST
1612 .ENDM PASS

```

```

1614      .SBTTL GLOBAL SUBROUTINE                - PREG05
1615      ;*****
1616      ;*      PRESERVE REGISTERS R0 THROUGH R5 FOR SUBROUTINE CALLS.
1617      ;*
1618      ;* INPUTS:      THE RETURN ADDRESS BACK INTO THE CALLING ROUTINE MUST BE IN
1619      ;*              GPR R5. (I.E. - MACROS USE "JSR R5,PREG05".)
1620      ;*
1621      ;* OUTPUTS:     REGISTERS R0 THROUGH R5 ARE SAVED ON THE STACK.
1622      ;*
1623      ;* CALLING SEQUENCE:  SAVE          ;MACRO EXPANSION CALLS PREG05.
1624      ;*                  [SUBROUTINE CODE]...
1625      ;*                  PASS          ;MACRO EXPANSION RECALLS PREG05.
1626      ;*
1627      ;* COMMENTS:      THIS ROUTINE IS RE-ENTRANT.
1628      ;*
1629      ;*                PARAMETERS MAY BE PASSED OUT OF A SUBROUTINE BY MODIFYING THE
1630      ;*                REGISTER SAVE AREA ON THE STACK.  USE THE PASS GPRN MACRO
1631      ;*                TO RETURN GPR VALUES INTACT.
1632      ;*                USE THE RNSLOT OFFSETS FROM THE SP TO PASS OTHER PARAMETERS.
1633      ;*                [EXAMPLE:      MOV    VALUE,R0SLOT(SP)      ]
1634      ;*                MAKE SURE THE SP IS AT ITS ORIGINAL VALUE WHEN YOU DO THIS.
1635      ;*
1636      ;* SUBORDINATE ROUTINES CALLED:  NONE.
1637      ;*****
1638
1639      PREG05:      ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
1640      MOV          R4,-(SP)      ;SAVE R4
1641      MOV          R3,-(SP)      ;SAVE R3
1642      MOV          R2,-(SP)      ;SAVE R2
1643      MOV          R1,-(SP)      ;SAVE R1
1644      MOV          R0,-(SP)      ;SAVE R0
1645      MOI          R5,-(SP)      ;PUSH RETURN PC ON TOP OF STACK
1646      MOV          R5SLOT(SP),R5 ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS
1647
1648      JSR          PC,0(SP)      ;CALL THE SUBROUTINE AT THE RETURN ADDRESS
1649      ;FROM THE PREG05 CALL, PUTTING THE PRESENT
1650      ;PC ON THE STACK AS A RETURN ADDRESS INTO
1651      ;THIS (PREG05) ROUTINE.
1652
1653      ;***
1654      ;THE FOLLOWING CODE IS EXECUTED WHEN THE CALLING ROUTINE DOES A
1655      ;"RETURN" (JSR PC,0(SP).) USING THE PC DEPOSITED ON THE STACK ABOVE.
1656      ;***
1657
1658      PREGRT:      MOV          (SP)+,R5      ;PUT RETURN PC IN R5.
1659      MOV          (SP)+,R0      ;RESTORE R0.
1660      MOV          (SP)+,R1      ;RESTORE R1.
1661      MOV          (SP)+,R2      ;RESTORE R2.
1662      MOV          (SP)+,R3      ;RESTORE R3.
1663      MOV          (SP)+,R4      ;RESTORE R4.
1664
1665      RTS          R5              ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05.
1666      ;RESTORING R5 IN THE PROCESS.

```

C4

GZDHXPO.DHU-11.FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 34

SEQ 41

1668	.SBTTL GLOBAL TEXT SECTION				
1676					
1677					
1678					
1679	; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,				
1680	; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN				
1681	; MORE THAN ONE TEST.				
1682	; --				
1683					
1684	; NAMES OF DEVICES SUPPORTED BY PROGRAM				
1685	;				
1686	DEV TYP <DHU-11>				
1687	004120				L\$DVTYP::
	004120				.ASCIZ /DHU-11/
	004120	104	110	125	
	004123	055	061	061	
	004126	000			.EVEN
1688					
1694					
1695	; TEST DESCRIPTION				
1696	;				
1697	004130				L\$DESC::
	004130				.ASCIZ /DHU-11 FUNC TST PAR
	004130	104	110	125	
T2/	004133	055	061	061	
	004136	040	106	125	
	004141	116	103	040	
	004144	124	123	124	
	004147	040	120	101	
	004152	122	124	062	
	004155	000			.EVEN
1698					
1699	.EVEN				
1706					

```

1715
1716      ,NLIST BIN
1717
1718
1719      ; ***** FORMAT STATEMENTS USED IN PRINT CALLS *****
1720
1721
1722 004156 EF0503:: .ASCIZ /#T#N/
1723 004163 EF1601:: .ASCIZ /#A #T#A, TEST ABORTED #N/
1724 004215 EF5801:: .ASCIZ /#A      RXTIMER VALUE USED WAS :#D3#A (D)#N/
1725 004270 EF5901:: .ASCIZ /#A      EXPECTED :#D3#A(D)#N/
1726 004320 EF5902:: .ASCIZ /#A      ACTUAL   :#D3#A(D)#N/
1727 004350 EF6401:: .ASCIZ /#A                        #D2#N/
1728 004417 EF7801:: .ASCIZ /#T#A ON LINE #D2#A DECIMAL.#N/
1729 004455 EF9001:: .ASCIZ /#A UNEXPECTED #T#A FOUND IN RECEIVE CHAR FIFO:#N/
1730 004537 EF9002:: .ASCIZ /#A      CODE IS ASSOCIATED WITH LINE: #D2#N/
1731 004611 EF9003:: .ASCIZ /#A      CODE IS: #D3#N/
1732 004640 EF9004:: .ASCIZ /#A      #T#A VALUE: #D3#N/
1733 004670 EF9005:: .ASCIZ /#A      #T#A VALUE: NONE#N/
1734 004721 EF9006:: .ASCIZ /#A #T#A #D2#A(D)#N/
1735 004745 EF9010:: .ASCIZ /#A      NUMBER OF ERRORS DETECTED ON LINE #D2#A IS #D5#N/
1736 005034 EF9019:: .ASCIZ /#A #T#A #D6#N/
1737 005053 EF9301:: .ASCIZ /#A #T#D2#A(D), BMP CODE REPORTED :#D3#A(O)#N/
1738 005131 EF9302:: .ASCIZ /#A OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)#N/
1739 005231 MFUNIT:: .ASCIZ /#N#A TESTING UNIT :#D4#N/
1740      .EVEN
1741      .LIST BIN

```

```

1750
1751      .NLIST BIN
1752
1753
1754      ;***** GLOBAL ERROR MESSAGES *****
1755
1756 005262 EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
1757 005320 EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
1758 005403 EM4001:: .ASCIZ /DMA_START BIT TEST FAILED/
1759 005435 EM4002:: .ASCIZ /DMA_START BIT BAD ON LINE: /
1760 005471 EM4101:: .ASCIZ /DMA_ABORT BIT TEST FAILED/
1761 005523 EM4102:: .ASCIZ /DMA_ABORT BIT BAD ON LINE: /
1762 005557 EM4103:: .ASCIZ /DMA_START BIT FOUND SET AFTER DMA ABORTED ON LINE: /
1763 005643 EM4201:: .ASCIZ /DMA_ERROR BIT TEST FAILED/
1764 005675 EM4202:: .ASCIZ /DMA_ERROR BIT BAD/
1765 005717 EM4901:: .ASCIZ /OAUTO (INACTIVE) BIT TEST FAILED/
1766 005760 EM4902:: .ASCIZ /OAUTO BIT BAD ON LINE: /
1767 006012 EM5001:: .ASCIZ /OAUTO (ACTIVE) BIT TEST FAILED/
1768 006051 EM5101:: .ASCIZ /IAUTO (INACTIVE) TEST FAILED/
1769 006106 EM5102:: .ASCIZ /IAUTO BIT FOUND SET ON LINE: /
1770 006144 EM5103:: .ASCIZ /IAUTO BIT BAD ON LINE: /
1771 006174 EM5201:: .ASCIZ /IAUTO (ACTIVE) TEST FAILED/
1772 006227 EM5202:: .ASCIZ /IAUTO BIT FOUND CLR ON LINE: /
1773 006265 EM5301:: .ASCIZ /FIFO VALID DATA TEST FAILED/
1774 006321 EM5302:: .ASCIZ /FIFO BAD, DATA FIELD CORRUPTED, TEST USED LINE:/
1775 006401 EM5303:: .ASCIZ /BMP CODE FOUND IN FIFO, TEST INVAILEDATED/
1776 006452 EM5401:: .ASCIZ /FIFO 3/4 ALARM (INACTIVE) TEST FAILED\
1777 006520 EM5402:: .ASCIZ /FIFO BAD, ALARM SIGNAL DEFECTIVE/
1778 006561 EM5501:: .ASCIZ /FIFO 3/4 ALARM (ACTIVE) TEST FAILED\
1779 006625 EM5601:: .ASCIZ /FIFO 3/4 ALARM (ACTIVE/INACTIVE) TEST FAILED\
1780 006702 EM5701:: .ASCIZ /FIFO 1/2 LEVEL (ACTIVE/INACTIVE) TEST FAILED\
1781 006757 EM5801:: .ASCIZ /RXTIMER TEST FAILED/
1782 007003 EM5802:: .ASCIZ /RXTIMER BAD, RX-INT DELAYED BY WRONG NUMBER OF MILLISECONDS/
1783 007077 EM5803:: .ASCIZ /RXTIMER BAD, RX-INT DIDN'T OCCUR IMMEDIATELY WITH RXFIFO 3/4 FULL\
1784 007201 EM5804:: .ASCIZ /RXTIMER BAD, RX-INT OCCURED WITH RXTIMER VALUE ZERO/
1785 007265 EM5805:: .ASCIZ /RXTIMER BAD, TIME-OUT OCCURED WAITING FOR RX-INT/
1786 007346 EM5901:: .ASCIZ /TX_ACTION FIFO TEST FAILED/
1787 007402 EM5902:: .ASCIZ /TX_ACTION FIFO BAD, TX_ACTION RECIEVED FROM THE WRONG LINE/
1788 007476 EM5903:: .ASCIZ /TX_ACTION FIFO BAD, INCORRECT NUMBER OF TX-ACTIONS FOUND/
1789 007570 EM5904:: .ASCIZ /TX_ACTION FIFO BAD, TX_ACTION FIFO WOULD NOT EMPTY/
1790 007654 EM5905:: .ASCIZ /TX_INTERRUPT OCCURED AFTER THE TX_ACTION FIFO HAD BEEN EMPTIED/
1791 007754 EM6001:: .ASCIZ /TX FIFO TEST FAILED/
1792 010000 EM6002:: .ASCIZ /INCORRECT VALUE IN FIFOSIZE REG/
1793 010040 EM6003:: .ASCIZ /MORE THAN ONE TX-INT OCCURED FROM A FULL TXFIFO/
1794 010120 EM6004:: .ASCIZ /TX FIFO BAD, RECIEVED CHAR INCORRECT/
1795 010165 EM6005:: .ASCIZ /TX FIFO BAD, CHARACTER RECIEVED ON WRONG LINE/
1796 010243 EM6006:: .ASCIZ /TX FIFO BAD, TOO FEW CHARS RECIEVED/
1797 010307 EM6401:: .ASCIZ /BREAK GENERATION TEST FAILED/
1798 010344 EM6402:: .ASCIZ /BREAK NOT RECEIVED ON LINE(S):/
1799 010405 EM6601:: .ASCIZ /NO OVERRUN ERROR TEST FAILED/
1800 010442 EM6602:: .ASCIZ /OVERRUN ERROR REPORTED WHEN NONE FORCED/
1801 010514 EM6701:: .ASCIZ /OVERRUN ERROR TEST FAILED/
1802 010546 EM6702:: .ASCIZ /NO OVERRUN ERROR REPORTED, OVERRUN FORCED/
1803 010623 EM9009:: .ASCIZ /EXPECTED OR CORRECT/
1804 010647 EM9010:: .ASCIZ /ACTUAL OR MEASURED /
1805 010673 EM9014:: .ASCIZ /SUMMARY REPORTS FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:/
1806 010767 EM9017:: .ASCII /FIFO WILL NOT PURGE (DATA_VALID STUCK SET),/

```

```
1807 011044 .ASCIZ / REMAINDER OF TEST SKIPPED./
1808 011100 EM9026:: .ASCIZ / LPR CONTENTS: /
1809 011124 EM9104:: .ASCIZ / UNEXPECTED DATA FOUND IN SIFO FROM LINE: /
1810 011200 EM9301:: .ASCIZ /BMP CODES WERE REPORTED DURING THIS DIAGNOSTIC/
1811 011257 EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
1812 011307 EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
1813 011354 EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
1814 .EVEN
1815 .LIST BIN
```

G4

CZDHVBO DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 37
GLOBAL TEXT SECTION

SEQ 45

1824
1825
1826
1827
1828
1829
1830
1831
1832

.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.
!--

```
1834 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0101 -
1835 *****
1836 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1837 ;* INFORMATION IF AN ERROR IS DETECTED IN TEST 1 (REGISTER ADDRESS
1838 ;* ACCESS TEST). IF THE "EXTENDED ERROR INFO" OPTION HAS BEEN SELECTED
1839 ;* THEN THIS SUBROUTINE WILL REPORT THE TYPE OF ACCESS (READ OR WRITE OR
1840 ;* BOTH) WHICH CAUSED A BUS TIME-OUT TRAP (004 TRAP). A MESSAGE INDICATING
1841 ;* THAT THE DHU MAY BE AT THE WRONG UNIBUS ADDRESS IS ALSO PRINTED.
1842 ;*
1843 ;* INPUTS: R5 - ERROR FLAG WORD.
1844 ;* IF BIT 0 IS SET, A READ ERROR OCCURED.
1845 ;* IF BIT 1 IS SET, A WRITE ERROR OCCURED.
1846 ;*
1847 ;* OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE.
1848 ;*
1849 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER0101" AS THE MESSAGE POINTER
1850 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
1851 ;*
1852 ;* COMMENTS:
1853 ;*
1854 ;* SUBORDINATE ROUTINES USED: NONE.
1855 *****
1856
1857 011430 .BGNMSG ER0101
1858 011430 ER0101::
1859 011430 SAVE JSR R5,PREG05 ;SAVE THE GPR CONTENTS.
1860 011434 012700 000100 ;SET-UP THE BIT MAP FOR 'REPORT EXT'D ERROR INFO'
1861 011440 046700 170544 ;TRY AND CLEAR THE FLAG.
1862 011444 001036 ;EXIT IF OPTION NOT SELECTED.
1863 ;*
1864 ;* REPORT EXTENDED ERROR INFORMATION
1865 ;*
1866
1867 011446 032705 000001 ;TEST FOR READ ERROR.
1868 011452 001410 ;SKIP READ ERROR MSG IF NO READ ERROR.
1869 011454 PRINTB MSG1 ;PRINT READ ERROR MESSAGE.
1870 011454 012746 011546 MOV MSG1, -(SP)
1871 011460 012746 000001 MOV #1, -(SP)
1872 011466 010600 MOV SP,R0
1873 011470 062706 000004 TRAP C$PNTB
1874 011474 032705 000002 ADD #4,SP
1875 011500 001410 ;TEST FOR WRITE ERROR.
1876 011502 PRINTB MSG2 ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
1877 011502 012746 011624 MOV MSG2, -(SP)
1878 011506 012746 000001 MOV #1, -(SP)
1879 011512 010600 MOV SP,R0
1880 011514 104414 TRAP C$PNTB
1881 011516 062706 000004 ADD #4,SP
1882 011522 PRINTX MSG3 ;SUGGEST THAT DHU MAY BE AT WRONG ADDRESS.
1883 011522 012746 011703 MOV MSG3, -(SP)
1884 011526 012746 000001 MOV #1, -(SP)
1885 011532 010600 MOV SP,R0
1886 011534 104415 TRAP C$PNTX
1887 011536 062706 000004 ADD #4,SP
```

CZDHV80 DHU-11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:28 PAGE 38-1
- ERO101 -

SEQ 47

```

1874 011542          64:  PASS          ;RESTORE THE GPR CONTENTS.
      011542 004736          JSR          PC,0(SP);RETURN TO PREG05 SUBRT.
1875 011544          ENDMSC
      011544          L10002:
      011544 104423          TRAP      C$MSG
1876
1877 011546          045      101      102  MSG1::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT,%N/
      011551          125      123      040
      011554          124      111      115
      011557          105      055      117
      011562          125      124      040
      011565          124      122      101
      011570          120      040      103
      011573          101      125      123
      011576          105      104      040
      011601          102      131      040
      011604          122      105      101
      011607          104      040      101
      011612          124      124      105
      011615          115      120      124
      011620          056      045      116
      011623          000
1878 011624          045      101      102  MSG2::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT,%N/
      011627          125      123      040
      011632          124      111      115
      011635          105      055      117
      011640          125      124      040
      011643          124      122      101
      011646          120      040      103
      011651          101      125      123
      011654          105      104      040
      011657          102      131      040
      011662          127      122      111
      011665          124      105      040
      011670          101      124      124
      011673          105      115      120
      011676          124      056      045
      011701          116      000
1879 011703          045      101      104  MSG3::  .ASCIZ  /%ADHU MAY BE AT THE WRONG UNIBUS ADDRESS,%N%N/
      011706          110      125      040
      011711          115      101      131
      011714          040      102      105
      011717          040      101      124
      011722          040      124      110
      011725          105      040      127
      011730          122      117      116
      011733          107      040      125
      011736          116      111      102
      011741          125      123      040
      011744          101      104      104
      011747          122      105      123
      011752          123      056      045
      011755          116      045      116
      011760          000

```

1880
1881

.EVEN

J4

CZDHV80 DHU-11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINEMACRO M1200 15-MAR-84 09:28 PAGE 39
- ER0503 -

SEQ 48

```

1883 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0503 -
1884 ;*****
1885 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
1886 ;* MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER, PROVIDED
1887 ;* EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
1888 ;*
1889 ;* INPUTS: R1 - ADDRESS OF THE MESSAGE TO PRINT.
1890 ;*
1891 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
1892 ;*
1893 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE MESSAGE IN R1.
1894 ;* INCLUDE THE LABEL "ER0503" AS THE MESSAGE POINTER
1895 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1896 ;*
1897 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
1898 ;*
1899 ;* SUBORDINATE ROUTINES USED: NONE.
1900 ;*****
1901
1902 011762 BGNMSG ER0503
1903 011762 ER0503::
1904 011762 012700 000100 MOV 0BIT06,R0 ;TRY TO CLEAR THE
1905 011766 046700 170216 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
1906 011772 001011 BNE 2$ ;EXIT IF FLAG NOT SET.
1907
1908
1909 011774 PRINTB 0EF0503,R1 ;PRINT THE MESSAGE.
1910 011774 010146 MOV R1,-(SP)
1911 011776 012746 004156 MOV 0EF0503,-(SP)
1912 012002 012746 000002 MOV 02,-(SP)
1913 012006 010600 MOV SP,R0
1914 012010 104414 TRAP C$PN'B
1915 012012 062706 000006 ADD 06,SP
1916
1917 012016 2$: ENDMSG
1918 012016 L10003.
1919 012016 104423 TRAP C$MSG

```

```

1913 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER1603 -
1914 ;*****
1915 ;* THIS ERROR REPORTING ROUTINE IS USED TO PRINT OUT A BASIC ERROR
1916 ;* MESSAGE, ALONG WITH A MESSAGE INFORMING THE OPERATOR WHICH TEST IS
1917 ;* ABOUT TO BE ABORTED, PROVIDED EXTENDED ERROR INFORMATION HAS BEEN
1918 ;* REQUESTED, OTHERWISE ONLY A "TEST FAILURE" MESSAGE WILL BE PRINTED.
1919 ;*
1920 ;* INPUTS: R1 - CONTAINS THE ADDRESS OF THE MESSAGE TO BE PRINTED.
1921 ;* ERRMSG - CONTAINS THE ADDRESS OF THE MESSAGE THAT INDICATES
1922 ;* THE TEST THAT IS BEING PERFORMED, EG DMA, BREAK ETC.
1923 ;*
1924 ;* OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
1925 ;* "TESTNAME TEST ABORTED"
1926 ;*
1927 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER1603" AS THE MESSAGE POINTER
1928 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
1929 ;*
1930 ;* COMMENTS:
1931 ;*
1932 ;*
1933 ;* SUBORDINATE ROUTINES CALLED: NONE.
1934 ;*****
1935 012020 BGNMSG ER1603
1936 012020 ER1603::
1937 012020 004567 172034 SAVE JSR ;SAVE THE CONTENTS OF THE GPRS.
1938 012024 012700 000100 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
1939 012030 046700 170154 MOV #BIT06,R0 ;TRY TO CLEAR THE
1940 012034 001024 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
1941 BNE 2$ ;EXIT IF FLAG NOT SET.
1942
1943 012036 PRINTB #EF0503,R1 ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
1944 012036 010146 MOV R1,-(SP)
1945 012040 012746 004156 MOV #EF0503,-(SP)
1946 012044 012746 000002 MOV #2,-(SP)
1947 012050 010600 MOV SP,R0
1948 012052 104414 TRAP C$PNTB
1949 012054 062706 000006 ADD #6,SP
1950
1951 012060 016702 171770 MOV ERRMSG,R2 ;GET THE "TEST MESSAGE".
1952 012064 PRINTB #EF1601,R2 ;PRINT "TEST ABORTED" MESSAGE.
1953 012064 010246 MOV R2,-(SP)
1954 012066 012746 004163 MOV #EF1601,-(SP)
1955 012072 012746 000002 MOV #2,-(SP)
1956 012076 010600 MOV SP,R0
1957 012100 104414 TRAP C$PNTB
1958 012102 062706 000006 ADD #6,SP
1959
1960 2$: PASS ;RESTORE THE CONTENTS OF THE GPRS.
1961 012106 004736 JSR PC,@(SP). ;RETURN TO PREG05 SUBRT.
1962 012110 ENDMSG
1963 012110 104423 L10004: TRAP C$MSG

```

```

1951      .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER5801 -
1952      ;*****
1953      ;* THIS IS AN ERROR REPORTING ROUTINE WHICH PRINTS THE MESSAGE PASSED
1954      ;* AS A PARAMETER IN R1, AND THE RXTIMER VALUE IN R2, PROVIDED
1955      ;* EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
1956      ;* THIS ROUTINE IS USED BY THE RXTIMER TEST.
1957      ;*
1958      ;* INPUTS:      R1 - ADDRESS OF THE MESSAGE TO PRINT.
1959      ;*              R2 - RXTIMER VALUE.
1960      ;*
1961      ;* OUTPUTS:     THE MESSAGE FOLLOWED BY THE RXTIMER VALUE ARE PRINTED AT
1962      ;*              THE OPERATOR CONSOLE.
1963      ;*
1964      ;* CALLING SEQUENCE:  INCLUDE THE LABEL ER5801 AS THE MESSAGE POINTER
1965      ;*                    PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
1966      ;*
1967      ;* COMMENTS:     THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION AND THE
1968      ;*              RXTIMER VALUE IS PRINTED AS A 3 DIGIT DECIMAL NUMBER.
1969      ;*
1970      ;* SUBORDINATE ROUTINES USED: NONE.
1971      ;*****
1972
1973      BGNMSG ER5801
1974      BIT 0BIT06,OPTION ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
1975      BEQ 2$           ;EXIT WITH "TEST FAILED" MESSAGE IF NOT.
1976
1977      PRINTB 0EF0503,R1
1978      PRINTB 0EF5801,R2
1979      2$:  ENDMSG

```

```

MOV R1,-(SP)
MOV 0EF0503,-(SP)
MOV 02,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD 06,SP

MOV R2,-(SP)
MOV 0EF5801,-(SP)
MOV 02,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD 06,SP

L10005: TRAP C$MSG

```

```

012112 012112
012112 032767 000100 170070
012120 001422
012122 010146
012124 012746 004156
012130 012746 000002
012134 010500
012136 104414
012140 062706 000006
012144 010246
012146 012746 004215
012152 012746 000002
012156 010600
012160 104414
012162 062706 000006
012166 104423

```

M4

CZDH/BO DHU-11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINEMACRO M1200 15-MAR-84 09:28 PAGE 42
- ER5901 -

SEQ 51

```

1981 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER5901 -
1982 ;*****
1983 ;* THIS IS AN ERROR REPORTING ROUTINE WHICH PRINTS AN ADDITIONAL
1984 ;* MESSAGE IN ADDITION TO THE "TEST FAILED" MESSAGE AND ALSO A
1985 ;* MESSAGE SHOWING THE EXPECTED VALUE OF A PIECE OF DATA AND THE
1986 ;* ACTUAL VALUE OF THAT DATA. THE DATA IS PRINTED AS A 3 DIGIT
1987 ;* DECIMAL NUMBER.
1988 ;*
1989 ;* INPUTS : R1 - EXPECTED VALUE OF DATA.
1990 ;* R2 - ACTUAL VALUE OF DATA.
1991 ;* R3 - ADDRESS OF THE MESSAGE TO PRINT.
1992 ;*
1993 ;* OUTPUTS : MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
1994 ;*
1995 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER5901" AS THE MESSAGE POINTER
1996 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
1997 ;*
1998 ;* COMMENTS :
1999 ;*
2000 ;* SUBORDINATE ROUTINES USED : NONE.
2001 ;*****
2002
2003 BGNMSG ER5901
2004 ER5901::
2005 SAVE ;SAVE THE GPR CONTENTS.
2006 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2007 BIT #BIT06,OPTION ;EXIT THE ROUTINE IF EXTENDED
2008 BEQ 60$ ;ERROR REPORTING IS NOT REQUESTED.
2009
2010 ;*
2011 ; REPORT EXTENDED ERROR INFORMATION
2012 ;*
2013
2014 PRINTB #EF0503,R3 ;PRINT THE MESSAGE.
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
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
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
435
```

N4

CZDHV80 DHU-11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:28 PAGE 42-1
- ER5901 -

SEQ 52

```
2018 012272          60$: PASS          ;RESTORE THE GPR CONTENTS.  
      012272 004736          JSR          PC,@(SP)+      ;RETURN TO PREG05 SUBRT.  
2019  
2020 012274          ENDMSG  
      012274  
      012274 104423          L10006: TRAP C$MSG
```

B5

CZDHVBO DMU-11 FUNC 1ST PART2
GLOBAL ERROR REPORTING ROUTINEMACRO M1200 15-MAR-84 09:28 PAGE 43
- ER6001 -

SEQ 53

```

2022 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER6001 -
2023 *****
2024 ;* THIS IS AN ERROR REPORTING ROUTINE WHICH PRINTS OUT A MESSAGE
2025 ;* AT THE CONSOLE INFORMING THE OPERATOR OF AN ERROR ON A PARTICULAR
2026 ;* LINE. THE ROUTINE ALSO PRINTS OUT A MESSAGE INFORMING THE OPERATOR
2027 ;* OF WHAT DATA WAS "EXPECTED" AND WHAT "ACTUAL" DATA WAS FOUND, IN THE
2028 ;* FORM OF A 3 DIGIT DECIMAL NUMBER.
2029 ;* IF EXTENDED ERROR REPORTING HAS NOT BEEN REQUESTED THEN ONLY THE
2030 ;* "TEST FAILED" MESSAGE WILL BE PRINTED.
2031 ;*
2032 ;* INPUTS : R1 - EXPECTED DATA
2033 ;* R2 - ACTUAL DATA
2034 ;* R3 - ADDRESS OF THE MESSAGE TO PRINT
2035 ;* R4 - LINE NUMBER ON WHICH THE ERROR OCCURED
2036 ;*
2037 ;* OUTPUTS : MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE
2038 ;* "MESSAGE" ON LINE 0"
2039 ;* "EXPECTED : " 00
2040 ;* "ACTUAL : " 00
2041 ;*
2042 ;* CALLING SEQUENCE : INCLUDE THE LABEL "ER6001" AS THE ERROR ROUTINE
2043 ;* POINTER PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
2044 ;*
2045 ;* SUBORDINATE ROUTINES CALLED : NONE
2046 ;* *****
2047
2048 012276 BGNMSG ER6001
2049 012276 ER6001::
2050 012276 032767 000100 167704 BIT #BIT06,OPTION ;EXIT THE ROUTINE IF EXTENDED
2051 012304 001434 604 REQ 604 ;ERROR REPORTING IS NOT REQUESTED.
2052 ;*
2053 ;* REPORT EXTENDED ERROR INFORMATION
2054 ;*
2055 012306 PRINTB 0EF7801,R3,R4 ;PRINT THE MESSAGE WITH THE LINE NUMBER.
2056 012306 010446 MOV R4,-(SP)
2057 012310 010346 MOV R3,-(SP)
2058 012312 012746 004417 MOV 0EF7801,-(SP)
2059 012316 012746 000003 MOV 03,-(SP)
2060 012322 010600 MOV SP,R0
2061 012324 104414 TRAP C:PNTB
2062 012326 062706 000010 ADD 01C,SP
2063 012332 PRINTX 0EF5901,R1 ;PRINT THE "EXPECTED" DATA MESSAGE.
2064 012332 010146 MOV R1,-(SP)
2065 012334 012746 004270 MOV 0EF5901,-(SP)
2066 012340 012746 000002 MOV 02,-(SP)
2067 012344 010600 MOV SP,R0
2068 012346 104415 TRAP C:PNTX
2069 012350 062706 000006 ADD 06,SP
2070 012354 PRINTX 0EF5902,R2 ;PRINT THE "ACTUAL" DATA MESSAGE.
2071 012354 010246 MOV R2,-(SP)
2072 012356 012746 004320 MOV 0EF5902,-(SP)
2073 012362 012746 000002 MOV 02,-(SP)
2074 012366 010600 MOV SP,R0
2075 012370 104415 TRAP C:PNTX
2076 012372 062706 000006 ADD 06,SP
2077
2078

```

CZDHVBO DHU 11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINE

C5
MACRO M1200 15-MAR-84 09:28 PAGE 43-1
- ER6001 -

SEQ 54

2059 012376
012376
012376 104423

604: ENDMSG

L10007: TRAP C4MSG

2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104

012400
012400
012400 004567 171454

012404 032767 000100 167576
012412 001433

012414 005002
012416 012703 000020
012422
012422 010146
012424 012746 004156
012430 012746 000002
012434 010600
012436 104414
012440 062706 000006
012444 000241
012446 006205
012450 103011
012452
012452 010246
012454 012746 004350
012460 012746 000002
012464 010600
012466 104414

SBTTL GLOBAL ERROR REPORTING ROUTINE - ER6401 -

THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
INFORMATION AFTER THE ERROR MESSAGE HEADER, PROVIDED EXTENDED ERROR
REPORTING HAS BEEN ENABLED.
THIS SUBROUTINE IS PASSED A CPR CONTAINING FLAGS WHICH INDICATE
THE LINE(S) FOR WHICH THE ERROR CONDITION SHOULD BE REPORTED.
INPUTS: R1 - ADDRESS OF THE MESSAGE TO BE PRINTED BY THIS ROUTINE.
R5 - CONTAINS THE ERROR FLAGS, (1 FLAG PER LINE).
OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE.
CALLING SEQUENCE: LOAD THE ADDRESS OF THE MESSAGE IN R1.
INCLUDE THE LABEL "ER6401" AS THE MESSAGE POINTER
PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
COMMENTS: THE OUTPUT FORMAT OF THIS MESSAGE IS:
TEXT MESSAGE
0NN
0NN
WHERE EACH "0NN" IS THE NUMBER OF A LINE WITH THE ERROR.
SUBORDINATE ROUTINES USED: NONE.

BGNMSG ER6401
ER6401::
SAVE JSR R5,PREG05 ;SAVE THE CONTENTS OF THE GPRS.
;CALL REGISTER SAVE SUBRT.
; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
BIT 0BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
BEQ 601 ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
;DURING THE SOFTWARE QUESTIONS.
CLR R2 ;CLEAR LINE NUMBER TO ZERO.
MOV 0NUMLNS,R3 ;SET UP MAX LINE COUNT.
PRINTB 0EF0503,R1 ;PRINT MESSAGE.
MOV R1, -(SP)
MOV 0EF0503, -(SP)
MOV 02, -(SP)
MOV SP,R0
TRAP C\$PNTB
ADD 06,SP
CLC ;CLEAR CARRY.
ASR R5 ;SHIFT FLAG OUT INTO CARRY BIT.
BCC 41 ;SKIP ERROR REPORT IF CLEAR.
PRINTB 0EF6401,R2 ;PRINT MESSAGE.
MOV R2, -(SP)
MOV 0EF6401, -(SP)
MOV 02, -(SP)
MOV SP,R0
TRAP C\$PNTB

E5

CZDHVBO DHU-11 FUNC 1ST PART2
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:28 PAGE 44-1
- ER6401 -

SEQ 56

ADDRESS	HEX	HEX	HEX	ASSEMBLY	COMMENT
2105	012470	062706	000006		ADD #6,SP
2106	012474	005202		4\$: INC R2	; INCREMENT LINE COUNT.
2107	012476	020302		CMP R3,R2	; CHECK IF MAX LINE COUNT EXCEEDED.
2108	012500	001362		BNE 2\$	; LOOP IF NOT DONE.
2109	012502			60\$: PASS	; RESTORE THE SAVED CONTENTS OF THE GPRS.
	012502	004736			PC,@(SP)+ ; RETURN TO PREG05 SUBRT.
	012504			ENDMSG	
	012504				L10010:
	012504	104423			TRAP C\$MSG

F5

CZD.V80 DHU-11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:28 PAGE 45
- ER7801 -

SEQ 57

```

2111      ,SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER7801 -
2112      ;*****
2113      ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
2114      ;*      MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER.  A LINE NUMBER
2115      ;*      IS INCLUDED AT THE END OF THE MESSAGE.  THE MESSAGE IS PRINTED ONLY IF
2116      ;*      EXTENDED ERROR REPORTING IS REQUESTED.
2117      ;*
2118      ;* INPUTS:      R1 - ADDRESS OF THE MESSAGE TO PRINT.
2119      ;*              R3 - NUMBER OF LINE ON WHICH ERROR OCCURRED.
2120      ;*
2121      ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
2122      ;*
2123      ;* CALLING SEQUENCE:  LOAD THE ADDRESS OF THE MESSAGE IN R1.
2124      ;*                  LOAD THE LINE NUMBER INTO R3.
2125      ;*                  INCLUDE THE LABEL "ER7801" AS THE MESSAGE POINTER
2126      ;*                  PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2127      ;*
2128      ;* COMMENTS:     THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2129      ;*
2130      ;* SUBORDINATE ROUTINES USED:  NONE.
2131      ;*****
2132
2133      BGNMSG  ER7801
2134
2135      ER7801::
2136
2137      ;*
2138      ;* EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
2139      ;*
2140      BIT     @BIT06,OPTION      ;EXIT WITH TEST FAILURE MESSAGE IF
2141      BEQ     2$                ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
2142                                ;DURING THE SOFTWARE QUESTIONS.
2143
2144      PRINTB  @EF7801,R1,R3      ;PRINT THE MESSAGE.
2145
2146      MOV     R3, -(SP)
2147      MOV     R1, -(SP)
2148      MOV     @EF7801, -(SP)
2149      MOV     R3, -(SP)
2150      MOV     SP,R0
2151      TRAP    C$PNTB
2152      ADD     @10,SP
2153
2154      2$:      ENDMSG
2155
2156      L10011:  TRAP    C$MSG

```

G5

CZDHVBO DHU-11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINEMACRO M1200 15-MAR-84 09:28 PAGE 46
- ER9001 -

SEQ 58

```

2146 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9001 -
2147 ;*****
2148 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH REPORTS AN UNEXPECTED
2149 ;* CODE WHICH HAS BEEN FOUND IN THE DUT CSR. THIS CODE CAN BE A BMP
2150 ;* CODE, A SELF-TEST CODE, OR A MODEM STATUS CODE.
2151 ;*
2152 ;* INPUTS: R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
2153 ;* R2 - SINGLE BYTE CODE WHICH HAS BEEN READ FROM THE DUT.
2154 ;* R4 - LINE NUMBER ASSOCIATED WITH THE CODE.
2155 ;*
2156 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
2157 ;*
2158 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9001" AS THE MESSAGE POINTER
2159 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2160 ;*
2161 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2162 ;*
2163 ;* SUBORDINATE ROUTINES USED: NONE.
2164 ;*****
2165
2166 012544 BGNMSG ER9001
2167 012544
2168
2169 ;*
2170 ;* EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
2171 ;*
2172 012544 032767 000100 167436 BIT #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
2173 012552 001433 BEQ 2$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
2174 ;* DURING THE SOFTWARE QUESTIONS.
2175 012554 PRINTB #EF9001,R1 ;REPORT TYPE OF CODE FOUND.
2176 012554 010146 MOV R1,-(SP)
2177 012556 012746 004455 MOV #EF9001,-(SP)
2178 012562 012746 000002 MOV #2,-(SP)
2179 012566 010600 MOV SP,R0
2180 012570 104414 TRAP C$PNTB
2181 012572 062706 000006 ADD #6,SP
2182 012576 PRINTX #EF9002,R4 ;REPORT THE LINE NUMBER OF THE CODE.
2183 012576 010446 MOV R4,-(SP)
2184 012600 012746 004537 MOV #EF9002,-(SP)
2185 012604 012746 000002 MOV #2,-(SP)
2186 012610 010600 MOV SP,R0
2187 012612 104415 TRAP C$PNTX
2188 012614 062706 000006 ADD #6,SP
2189 012620 PRINTX #EF9003,R2 ;REPORT THE CODE WHICH WAS FOUND.
2190 012620 010246 MOV R2,-(SP)
2191 012622 012746 004611 MOV #EF9003,-(SP)
2192 012626 012746 000002 MOV #2,-(SP)
2193 012632 010600 MOV SP,R0
2194 012634 104415 TRAP C$PNTX
2195 012636 062706 000006 ADD #6,SP
2196
2197 012642 2$: ENDMSG
2198 012642
2199 012642 104423 L10012: TRAP C$MSG

```

H5

CZDHV80 DHU-11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINEMACRO M1200 15-MAR-84 09:28 PAGE 47
- ER9002 -

SEQ 59

```

2181      ,SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9002 -
2182      ;*****
2183      ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS INTENDED FOR USE IN THE
2184      ;* TRANSMISSION AND RECEPTION TESTS. IT REPORTS THE TYPE OF ERROR WHICH
2185      ;* HAS OCCURRED WHEN INCORRECT DATA IS RECEIVED FROM THE DUT. THIS
2186      ;* ROUTINE ALSO REPORTS THE READ AND EXPECTED DATA VALUES.
2187      ;*
2188      ;* INPUTS:      R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
2189      ;*              R2 - DATA BYTE READ FROM THE DUT.
2190      ;*              R3 - LINE NUMBER MULTIPLIED BY 2.
2191      ;*              R4 - EXPECTED DATA BYTE, BIT 15 SET IF "NONE".
2192      ;*
2193      ;* OUTPUTS:     A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2194      ;*
2195      ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9002" AS THE MESSAGE POINTER
2196      ;*                   PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2197      ;*
2198      ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2199      ;*
2200      ;* SUBORDINATE ROUTINES USED: PRTLPR.
2201      ;*****
2202
2203      BGNMSG ER9002
2204
2205      ER9002::
2206
2207      ;*
2208      ;* EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
2209      ;*
2210      BIT    #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
2211      BEQ    62$           ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
2212                        ;DURING THE SOFTWARE QUESTIONS.
2213
2214      ASR    R3            ;CALCULATE THE LINE NUMBER.
2215      BIC    #177400,R2    ;MASK OUT ALL BUT DATA IN READ CHAR.
2216      PRINTB #EF9006,R1,R3 ;PRINT THE FIRST LINE OF THE MESSAGE.
2217
2218      MOV    R3, -(SP)
2219      MOV    R1, -(SP)
2220      MOV    #EF9006, -(SP)
2221      MOV    #3, -(SP)
2222      MOV    SP,R0
2223      TRAP   C$PNTB
2224      ADD    #10,SP
2225
2226      PRINTX #EF9004,#EM9010,R2 ;PRINT ACTUAL DATA.
2227
2228      MOV    R2, -(SP)
2229      MOV    #EM9010, -(SP)
2230      MOV    #EF9004, -(SP)
2231      MOV    #3, -(SP)
2232      MOV    SP,R0
2233      TRAP   C$PNTX
2234      ADD    #10,SP
2235
2236      TST    R4            ;CHECK FOR "NONE" CODE SET IN EXPECTED DATA.
2237      BMI    2$           ;BRANCH TO PRINT "NONE" MESSAGE IF FLAG SET.
2238      PRINTX #EF9004,#EM9009,R4 ;PRINT EXPECTED DATA.
2239
2240      MOV    R4, -(SP)
2241      MOV    #EM9009, -(SP)
2242      MOV    #EF9004, -(SP)
2243      MOV    #3, -(SP)

```

CZDHV80 DHU-11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 15-MAR-84 09:28 PAGE 47-1
- ER9002 -

SEQ 60

012756	010600					MOV	SP,RO
012760	104415					TRAP	C\$PNTX
012762	062706	000010				ADD	#10,SP
2219	012766	000412		BR	60\$	;EXIT THIS ROUTINE.	
2220	012770		2\$:	PRINTX	#EF9005,#EM9009	;PRINT MESSAGE INDICATING NO EXPECTED DATA.	
	012770	012746				MOV	#EM9009,-(SP)
	012774	012746				MOV	#EF9005,-(SP)
	013000	012746				MOV	#2,-(SP)
	013004	010600				MOV	SP,RO
	013006	104415				TRAP	C\$PNTX
	013010	062706				ADD	#6,SP
2221	013014	004767		60\$:	JSR	PC,PRTLPR	;PRINT CONTENTS OF THE LPR REGISTER.
2222	013020			62\$:	ENDMSG		
	013020					L10013:	
	013020	104423				TRAP	C\$MSG

```
2224 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9004 -
2225 ;*****
2226 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH REPORTS ERROR SUMMARIES
2227 ;* FOR LINES WHICH HAVE EXCEEDED THE SPECIFIED MAXIMUM NUMBER OF
2228 ;* INDIVIDUAL RECEPTION ERRORS, PROVIDED EXTENDED ERROR REPORTING HAS
2229 ;* BEEN REQUESTED BY THE OPERATOR.
2230 ;*
2231 ;* INPUTS: R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
2232 ;* ERCNTB - LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
2233 ;* ERSRNF - "REPORT ERROR SUMMARY FOR LINE" FLAGS.
2234 ;*
2235 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2236 ;*
2237 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9004" AS THE MESSAGE POINTER
2238 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2239 ;*
2240 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2241 ;* THE CONTENTS OF GPR'S R2, R3, R4, AND R5 ARE DESTROYED.
2242 ;*
2243 ;* SUBORDINATE ROUTINES USED: NONE.
2244 ;*****
2245
2246 013022 BGNMSG ER9004
2247 013022 ER9004::
2248 013022 012700 000100 MOV #BIT06,R0 ;TRY TO CLEAR THE
2249 013026 C46700 167156 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2250 013032 001040 BNE 6$ ;EXIT IF FLAG NOT SET.
2251 013034 PRINTB #EF0503,#EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
2252 013034 012746 010673 MOV #EM9014,-(SP)
2253 013040 012746 004156 MOV #EF0503,-(SP)
2254 013044 012746 000002 MOV #2,-(SP)
2255 013050 010600 MOV SP,R0
2256 013052 104414 TRAP C$PNTB
2257 013054 062706 000006 ADD #6,SP
2258 013060 005002 CLR R2 ;CLEAR THE LINE COUNTER.
2259 013062 016703 167524 MOV ERSRNF,R3 ;GET THE ERROR SUMMARY FLAGS.
2260 013066 005004 CLR R4 ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
2261 013070 000241 2$: CLC ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
2262 013072 006003 ROR R3 ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
2263 013074 103013 BCC 4$ ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
2264 013076 PRINTX #EF9010,R2,ERCNTB(R4)
2265 013076 016446 002614 MOV ERCNTB(R4),-(SP)
2266 013102 010246 MOV R2,-(SP)
2267 013104 012746 004745 MOV #EF9010,-(SP)
2268 013110 012746 000003 MOV #3,-(SP)
2269 013114 010600 MOV SP,R0
2270 013116 104415 TRAP C$PNTX
2271 013120 062706 000010 ADD #10,SP
2272 013124 012405 4$: MOV (R4)+,R5 ;INCREMENT THE LINE OFFSET BY 2.
2273 013126 005202 INC R2 ;INCREMENT THE LINE COUNTER.
2274 013130 005703 RST R3 ;CHECK THE ERROR SUMMARY FLAGS.
2275 013132 001356 BNE 2$ ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.
2276 013134 6$: ENDMMSG
2277 013134 104423 L10014: TRAP C$MSG
```

K5

CZDHV80 DHU-11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINEMACRO M1200 15-MAR-84 09:28 PAGE 49
- ER9101 -

SEQ 62

```

2264      ,SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9101 -
2265      ;*****
2266      ;*      THIS IS A GENERAL ERROR REPORTING SUBROUTINE WHICH REPORTS A MESSAGE
2267      ;*      WHICH TAKES A SINGLE, 2 DIGIT DECIMAL ARGUMENT AFTER THE END OF AN
2268      ;*      ASCII MESSAGE.
2269      ;*
2270      ;* INPUTS:      R1 - VALUE TO BE PRINTED AFTER MSG AS 2 DECIMAL DIGITS.
2271      ;*              R2 - ADDRESS OF MESSAGE TO PRINT FIRST.
2272      ;*
2273      ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
2274      ;*
2275      ;* CALLING SEQUENCE:  INCLUDE THE LABEL "ER9101" AS THE MESSAGE POINTER
2276      ;*                   PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2277      ;*
2278      ;* COMMENTS:     THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2279      ;*
2280      ;* SUBORDINATE ROUTINES USED: NONE.
2281      ;*****
2282
2283 013136      BGNMSG  ER9101
2284 013136
2285
2285 013136 012700 000100      MOV  08BIT06,R0      ;TRY TO CLEAR THE
2286 013142 046700 167042      BIC  OPTION,R0      ;EXT'D ERROR REPORTING FLAG
2287 013146 001012      BNE  2$      ;EXIT IF FLAG NOT SET.
2288
2289
2290 013150      PRINTB  0EF9006,R2,R1      ;REPORT THE STRING FOLLOWED BY THE NUMBER.
2291 013150 010146      MOV  R1,-(SP)
2292 013152 010246      MOV  R2,-(SP)
2293 013154 012746 004721      MOV  0EF9006,-(SP)
2294 013160 012746 000003      MOV  03,-(SP)
2295 013164 010600      MOV  SP,R0
2296 013166 104414      TRAP  C$PNTB
2297 013170 062706 000010      ADD  010,SP
2298
2299
2299 013174      2$:  ENDMSG
2299 013174
2299 013174 104423      L10015:  TRAP  C$MSG

```

```

2294 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9301 -
2295 ;*****
2296 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ANY BMP CODES
2297 ;* THAT ARE FOUND IN THE BMP CODE QUEUE, TOGETHER WITH THE THE NUMBER OF
2298 ;* THE TEST THAT WAS EXECUTING AT THE TIME THE BMP CODE WAS LOGGED.
2299 ;* PROVIDED EXTENDED ERROR REPORTING HAS BEEN ENABLED.
2300 ;*
2301 ;* INPUTS: R1 - THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
2302 ;* R2 - THE ADDRESS OF THE NEXT EMPTY CELL IN THE QUEUE.
2303 ;*
2304 ;* OUTPUTS: THE TEST NUMBER FOLLOWED BY THE BMP CODE ARE PRINTED AT THE
2305 ;* OPERATOR CONSOLE.
2306 ;*
2307 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9301" AS THE MESSAGE POINTER
2308 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2309 ;*
2310 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2311 ;*
2312 ;* SUBORDINATE ROUTINES USED: NONE.
2313 ;*****
2314
2315 013176 BGNMSG ER9301
2316 013176 ER9301::
2317 013176 004567 170656 SAVE JSR R5,PREG05 ;SAVE THE GPRS ON THE STACK.
2318 013202 012700 000100 ;CALL REGISTER SAVE SUBRT.
2319 013206 046700 166776 MOV #BIT06,R0 ;TRY TO CLEAR THE
2320 013212 001064 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2321 ;EXIT IF FLAG NOT SET.
2322 013214 PRINTB #EF0503,R1 ;REPORT UNEXPECTED BMP CODES FOUND.
2323 013214 010146 MOV R1,-(SP)
2324 013216 012746 004156 MOV #EF0503,-(SP)
2325 013222 012746 000002 MOV #2,-(SP)
2326 013226 010600 MOV SP,R0
2327 013230 104414 TRAP C$PNTB
2328 013232 062706 000006 ADD #6,SP
2329 013236 012703 002412 MOV #BMPCQ8,R3 ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
2330 013242 012705 011257 MOV #EM9302,R5 ;GET THE MESSAGE TO BE REPORTED.
2331 2$: MOV (R3)+,R1 ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
2332 MOV (R3)+,R4 ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
2333 JSR PC,50$ ;GO REPORT THE BMP CODE.
2334 CMP R3,R2 ;CHECK IF ALL CODES HAVE BEEN REPORTED.
2335 BLO 2$ ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
2336 ;
2337 ; CHECK IF OVERFLOW HAS OCCURRED.
2338 ; THE CONDITIONS FOR OVERFLOW ARE: THE POINTER CONTAINS THE ADDRESS OF THE
2339 ; LAST CELL IN THE QUEUE, AND A BMP CODE HAS ALREADY BEEN WRITTEN INTO THAT
2340 ; CELL.
2341 ;
2342 013262 020227 002606 CMP R2,#BMPCQE-4 ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
2343 013266 001036 BNE 60$ ;EXIT IF NOT AT THE LAST LOCATION.
2344 013270 005762 000002 IST 2(R2) ;CHECK FOR A BMP CODE IN THE LAST CELL
2345 013274 001433 BEQ 60$ ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
2346 013276 012301 MOV (R3)+,R1 ;GET THE TEST NUMBER OFF THE QUEUE.
2347 013300 011304 MOV (R3),R4 ;GET THE BMP CODE OFF THE QUEUE.
2348 013302 012705 011307 MOV #EM9303,R5 ;SELECT THE MESSAGE TO BE REPORTED.

```

M5

CZDHVBO DHU-11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINEMACRO M1200 15-MAR-84 09:28 PAGE 50-1
- ER9301 -

SEQ 64

```

2343 013306          PRINTX  #EF9302          ;REPORT OVERFLOW CONDITION.
      013306 012746 005131
      013312 012746 000001
      013316 010600
      013320 104415
      013322 062706 000004
2344 013326 004767 000002          JSR      PC,50$          ;REPORT THE LAST BMP CODE PLACED ON THE
2345 013332 000414          BR      60$          ;EXIT.
2346
2347 013334          50$: PRINTX  #EF9301,R5,R1,R4  ;PRINT THE MESSAGE.
      013334 010446
      013336 010146
      013340 010546
      013342 012746 005053
      013346 012746 000004
      013352 010600
      013354 104415
      013356 062706 000012
2348 013362 000207
2349 013364          60$: RTS      PC          ;RETURN.
      013364 004736          PASS          ;RESTORE THE GPR CONTENTS.
2350
2351 013366          JSR      PC,@(SP)+      ;RETURN TO PREG05 SUBRT.
      013366
      013366 104423          L10016: TRAP    C$MSG

```

N5

CZDHVBO DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 51
GLOBAL SUBROUTINES SECTION

SEQ 65

2353
2361
2362
2363
2364
2365
2366

.SBTTL GLOBAL SUBROUTINES SECTION

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

B6

CZDHV80 DMU-11 FUNC TST PART2
GLOBAL SUBROUTINEMACRO M1200 15-MAR-84 09:28 PAGE 52
- ALTFLD -

SEQ 66

```

2368 .SBTTL GLOBAL SUBROUTINE - ALTFLD -
2369 *** *****
2370 * - ALTER DEVICE REGISTER FIELDS ROUTINE -
2371 * THIS SUBROUTINE ALTERS THE SPECIFIED FIELD OF THE SPECIFIED DEVICE
2372 * REGISTER FOR THE SPECIFIED LINES. THIS ROUTINE CAN BE USED TO SET
2373 * OR CLEAR BITS WITHIN SELECTED FIELDS OF SELECTED REGISTERS.
2374 * USE EXAMPLES: SET RX,BAUD,RATE FIELDS ON LINES 3 AND 6.
2375 * CLEAR TX,DMA BITS ON ALL LINES.
2376 *
2377 * INPUTS: R1 - ADDRESS OF THE REGISTERS TO ALTER.
2378 * R2 - BIT FIELDS SET TO DESIRED STATES.
2379 * R3 - BIT MAP OF LINES FOR WHICH TO ALTER REGISTER.
2380 * R4 - MASK OF BITS TO ALTER (1 INDICATES CHANGE BIT).
2381 * CSRA - CONTAINS THE ADDRESS OF THE DEVICE CSR.
2382 * IESTAT - SAVED STATES OF THE INTERRUPT ENABLE BITS.
2383 *
2384 * OUTPUTS: DEVICE REGISTERS - SPECIFIED REGISTER FIELDS ALTERED.
2385 * CSR IND.ADR.REG FIELD - DESTROYED.
2386 *
2387 * CALLING SEQUENCE: JSR PC,ALTFLD
2388 *
2389 * COMMENTS: THIS ROUTINE READS THE SPECIFIED REGISTERS FOR ALL LINES
2390 * WITH NUMBERS LOWER THAN THE HIGHEST SPECIFIED LINE.
2391 * THIS ROUTINE DOES NOT READ THE CSR.
2392 *
2393 * SUBROUTINES CALLED: NONE.
2394 * - - - *****
2395
2396 013370 ALTFLD:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
013370 004567 170464 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2397
2398
2399 * SET UP TO LOOP FOR EACH LINE:
2400 * PREPARE THE WORD TO BE ORED INTO THE REGISTER CONTENTS.
2401 * SET UP THE WORD TO WRITE INTO THE IND.ADR.REG FIELD OF THE CSR.
2402
2403 013374 010100 MOV R4,R0 ;CALCULATE THE NEW CONTENTS OF THE
2404 013376 005100 COM R0 ; REGISTER FIELDS WHICH ARE TO BE
2405 013400 040002 BIC R0,R2 ; ALTERED BY THIS ROUTINE.
2406 013402 016705 166650 MOV IESTAT,R5 ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.
2407
2408 * LOOP ONCE FOR EACH LINE, ALTERING THE SPECIFIED FIELD IN THE SPECIFIED
2409 * REGISTER IF THE LINE HAS BEEN SELECTED FOR ALTERING.
2410 * EXIT THE LOOP IF NO MORE LINES TO ALTER, OR IF WE HAVE ALTERED THE MAX
2411 * ALLOWABLE NUMBER OF LINES (AS SPECIFIED BY NUMLNS).
2412
2413 013406 000241 CLC ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
2414 013410 006003 ROR R3 ;GET THE LINE SELECT BIT FOR THIS LINE.
2415 013412 103006 BCC 41 ;SKIP SETUP IF LINE IS NOT SELECTED.
2416 013414 010577 MOV R5,DCSRA ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
2417 013420 011100 MOV (R1),R0 ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
2418 013422 040400 BIC R4,R0 ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
2419 013424 050200 BIS R2,R0 ;OR IN THE NEW STATES OF THE FIELDS.
2420 013426 010011 MOV R0,(R1) ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
2421 013430 005205 41: INC R5 ;SET LINE NUMBER TO THE NEXT LINE.
2422 013432 005703 TST R3 ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
2423 013434 001365 BNE 21 ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.

```

C6
CZDHVRO DHU 11 LUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 52-1
GLOBAL SOBROUTINE ACTFLD

SEQ 67

2424							
2425	013436		603:	PASS		,RESTORE GPRS.	
	013436	004736			JSR	PC,8(SP).	,RETURN TO PREG05 SUBRT.
2426	013440	000207		RTS	PC		,RETURN TO CALLING ROUTINE.

D6

CZDHVBO DMU 11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 53
GLOBAL SUBROUTINE - ASLNTL -

SEQ 68

```

2428 .SBTTL GLOBAL SUBROUTINE - ASLNTL -
2429 ;* *****
2430 ;* - SETUP ASSOCIATED LINE NUMBER TABLES ROUTINE -
2431 ;* THIS ROUTINE SETS UP THE TWO TABLES WHICH ARE CONTAIN INFORMATION
2432 ;* ABOUT THE TX/RX LINE WHICH IS ASSOCIATED WITH A PARTICULAR RX/TX
2433 ;* LINE. ONE TABLE IS A TABLE OF WORDS WHICH CONTAINS WORD OFFSET
2434 ;* VALUES AND THE OTHER TABLE IS A TABLE OF BYTES WHICH CONTAINS
2435 ;* LINE NUMBER VALUES.
2436 ;*
2437 ;* INPUTS: LOPBCK - STORAGE FOR THE TYPE OF LOOPBACK ON THE OUT.
2438 ;* NUMLNS - EQUATED TO THE NUMBER OF LINES ON THE OUT.
2439 ;* STGTRB - LABEL AT BASE OF STAGGERED LINE ASSOCIATION TBL.
2440 ;* TXRLNB - LABEL AT BASE OF BYTE TX/RX LINE NUMBER TABLE.
2441 ;* TXRXLB - LABEL AT BASE OF WORD TX/RX LINE NUMBER TABLE.
2442 ;* TXRXLE - LABEL AT END OF WORD TX/RX LINE NUMBER TABLE.
2443 ;*
2444 ;* OUTPUTS: TXRXLB, TXRLNB - TABLES INITIALIZED FOR SELECTED LOOPBACK.
2445 ;*
2446 ;* CALLING SEQUENCE: JSR PC,ASLNTL
2447 ;*
2448 ;* COMMENTS:
2449 ;*
2450 ;* SUBORDINATE ROUTINES CALLED: NONE.
2451 ;* - *****
2452
2453 013442 ASLNTL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2454 013442 004567 170412 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2455 013446 126727 166550 000002 CMPB LOPBCK,02 ;TEST FOR STAGGERED LOOPBACK.
2456 013454 001411 BEQ 4$ ;GO SET UP STAGGERED TABLE IF STAGGERED LPBCK.
2457 ;*
2458 ;* SET UP THE WORD TABLE FOR NON-STAGGERED LOOPBACK.
2459 013456 005005 CLR R5 ;CLEAR THE LINE COUNTER
2460 013460 010565 003750 2$: MOV R5, TXRXLB(R5) ;SET UP A WORD OF THE TABLE.
2461 013464 005205 INC R5
2462 013466 005205 INC R5 ;SET LINE COUNTER TO NEXT LINE OFFSET.
2463 013470 020527 000040 CMP R5,02*NUMLNS ;TEST FOR ALL LINES DONE.
2464 013474 002771 BLT 2$ ;LOOP UNTIL ALL LINES DONE.
2465 013476 000411 BR 8$ ;GO SET UP THE BYTE TABLE.
2466 ;*
2467 ;* SET UP THE WORD TABLE FOR STAGGERED LOOPBACK.
2468 ;*
2469 013500 012701 004030 4$: MOV 0STGTRB,R1 ;SET UP THE SOURCE POINTER.
2470 013504 012702 003750 MOV 0TXRXLB,R2 ;SET UP THE DESTINATION POINTER.
2471 013510 112122 6$: MOVB (R1), (R2) ;MOVE A BYTE INTO THE TABLE.
2472 013512 105022 CLRB (R2) ;CLEAR THE UPPER BYTE OF WORD TABLE ENTRY.
2473 013514 020227 004010 CMP R2,0TXRXLE ;COMPARE POINTER WITH END ADR OF TABLE.
2474 013520 002773 BLT 6$ ;LOOP IF NOT AT END YET.
2475 ;*
2476 ;* SET UP THE BYTE TABLE BASED ON THE WORD ASSOCIATION TABLE.
2477 ;*
2478 013522 012701 003750 8$: MOV 0TXRXLB,R1 ;SET UP THE SOURCE POINTER.
2479 013526 012702 004010 MOV 0TXRLNB,R2 ;SET UP THE DESTINATION POINTER.
2480 013532 012103 10$: MOV (R1), R3 ;GET THE WORD OFFSET VALUE FROM WORD TABLE.
2481 013534 006203 ASR R3 ;DIVIDE BY 2 TO GET LINE NUMBER VALUE.
2482 013536 110322 MOVB R3, (R2) ;LOAD THE BYTE LINE NUMBER INTO TABLE.
2483 013540 020127 004010 CMP R1,0TXRXLE ;COMPARE SOURCE POINTER WITH ADR OF TABLE END.

```

MACRO M1200
- ASLNTL -

46

2484 013544 002772
2485
2486 013546
013546 004736
2487 013550 000207

	BLT	10\$
60\$:	PASS	
	RTS	PC

```

;LOOP IF NOT AT END OF TABLE YET.
;RESTORE GPRS.
JSR PC,0(SP)+ ;RETURN TO PREG05 SUBRT.

```

```
2489 .SBTTL GLOBAL SUBROUTINE - CALMSL -
2490 ;* *****
2491 ;* - CALIBRATE MILLI SECOND LOOP COUNT SUBROUTINE -
2492 ;* THIS SUBROUTINE CALIBRATES THE TIMING LOOP WHICH IS USED IN THE MSLOOP
2493 ;* ROUTINE. THIS SUBROUTINE CALCULATES A VALUE FOR THE MSLCNT VARIABLE
2494 ;* WHICH IS THE NUMBER OF SOFTWARE LOOPS WHICH TAKES 1 MS TO EXECUTE IN
2495 ;* THE MSLOOP ROUTINE. THIS ROUTINE CALIBRATES THE COUNT BY USING THE
2496 ;* LINE TIME CLOCK (LTC), SO IF NO LTC IS AVAILABLE THE DEFAULT VALUE FOR
2497 ;* THE DELAY COUNT MUST BE USED.
2498 ;*
2499 ;*
2500 ;* INPUTS: MSLCNT - DEFAULT 1 MS DELAY LOOP COUNT VALUE, OR
2501 ;* VALUE FROM PREVIOUS CALIBRATION.
2502 ;* MSTICK - NUMBER OF MS PER LTC CLOCK TICK.
2503 ;* TIMER1 - TIMER COUNTER CHANGED BY LTC INTERRUPT SERVICE RTN.
2504 ;* CLKHRZ - NUMBER OF LTC CLICKS PER SECOND (50 OR 60).
2505 ;*
2506 ;* OUTPUTS: CARRY - SET IF LTC IS AVAILABLE, AND NEW CALIBRATION PERFORMED.
2507 ;* MSLCNT - NEW 1 MS DELAY LOOP COUNT VALUE IF LTC AVAILABLE, OR
2508 ;* UNCHANGED IF NO LTC IS AVAILABLE.
2509 ;*
2510 ;* CALLING SEQUENCE: JSR PC,CALMSL
2511 ;*
2512 ;* COMMENTS:
2513 ;*
2514 ;* SUBORDINATE ROUTINES CALLED: UNSDIV, OOPS.
2515 ;* *****
2516
2517 013552 CALMSL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2518 013552 004567 170302 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2519 013556 005067 000210 CLR 62$ ;CLEAR THE 2ND TIME FLAG.
2520 ;*
2521 ;* SYNCHRONIZE WITH THE LTC.
2522 013562 012705 000001 2$: MOV 01,R5 ;SET OUTER LOOP COUNTER TO 1 LOOP.
2523 ;* ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE <*&
2524 ;* FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <*&
2525 013566 005000 CLR R0 ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
2526 013570 012767 000001 166516 MOV 01,TIMER1 ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
2527 013576 005767 166512 4$: TST TIMER1 ;CHECK FOR COUNTER HAVING GONE TO ZERO.
2528 013602 001410 BEQ 6$ ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
2529 013604 005200 INC R0 ;COUNT THIS ITERATION OF THE INNER LOOP.
2530 013606 001373 BNE 4$ ;LOOP IF COUNTER HAS NOT TURNED OVER.
2531 013610 005305 DEC R5 ;DECREMENT THE INNER LOOP COUNTER.
2532 013612 003371 BGT 4$ ;LOOP IF OUTER LOOP COUNT NOT UP.
2533 ;*
2534 ;* IF WE GOT NO LTC INTERRUPT, INDICATE THAT THERE IS NO LTC AVAILABLE.
2535 ;* LTC MUST BE FLAKEY, OR NOT REALLY AN LTC AT ALL.
2536 ;*
2537 013614 005067 166472 CLR CLKHRZ ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
2538 013620 000241 CLC ;INDICATE FAILURE FOR RETURN.
2539 013622 000461 BR 60$ ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
2540 ;*
2541 ;* WE ARE NOW SYNCHRONIZED WITH THE LTC.
2542 ;* SET UP FOR THE CALIBRATION LOOP.
2543 ;*
2544 013624 012704 002314 6$: MOV 01,TIMER1,R4 ;WILL TEST TIMER1 IN THE LOOP BELOW.
```

```

2543 013630 005001          CLR      R1          ;CLEAR THE OUTER LOOP COUNTER.
2546 013632 005002          CLR      R2          ;INDICATE TO CHECK ALL BITS OF TIMER1.
2547 013634 005003          CLR      R3          ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2548 013636 012714 000001    MOV      R1,(R4)    ;LOAD TIMER1 WITH COUNT OF 1.
2549
2550 013642 016705 166460    8$:      MOV      MSLCNT,R5    ;LOAD MS LOOP COUNT.
2551 013646 011400          10$:     MOV      (R4),R0    ;GET THE TIMER1 VALUE.
2552 013650 010067 000120    MOV      R0,64$    ;SAVE WORD (LIKE IN THE REAL LOOP).
2553 013654 040200          BIC      R2,R0    ;LEAVE ALL THE BITS.
2554 013656 020003          CMP      R0,R3    ;COMPARE AGAINST ZERO.
2555 013660 000261          SEC            ;SET CARRY IN CASE OF SUCCESS.
2556 013662 001406          BEQ      12$    ;EXIT LOOP IF TIMER1 HAS CLEARED.
2557 013664 005305          DEC      R5    ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2558 013666 001367          BNE      10$    ;LOOP IF MS NOT UP.
2559 013670 005301          DEC      R1    ;DECREMENT THE MS TIME COUNT.
2560 013672 001363          BNE      8$    ;KEEP LOOPING.
2561 013674 004767 001160    JSR      PC,OOPS    ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
2562
2563      ;*
2564      ; WE HAVE NOW HAVE LOOP COUNT INFORMATION FOR ONE CLOCK TICK.
2565      ; WE HAVE NEGATIVE OF NUMBER OF OUTER LOOPS IN R1, EACH IS MSLCNT INNER LOOPS.
2566      ; WE HAVE THE PORTION OF THE LAST OUTER LOOP NOT EXECUTED, IN R5.
2567      ; NOW WE CALCULATE THE TOTAL NUMBER OF INNER LOOPS EXECUTED.
2568 013700 005401          12$:     NEG      R1          ;GET NUMBER OF OUTER LOOPS.
2569 013702 016702 166420    MOV      MSLCNT,R2    ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2570 013706 010203          MOV      R2,R3    ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2571 013710 160502          SUB      R5,R2    ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2572 013712 010204          MOV      R2,R4    ; AND ADD TO ACCUMULATOR LSWORD.
2573 013714 005005          CLR      R5    ;CLEAR ACCUMULATOR MSWORD.
2574 013716 005301          14$:     DEC      R1    ;CHECK R1 FOR 0 CONDITION
2575 013720 100403          BMI      16$    ; SKIP MULTIPLICATION IF ZERO
2576 013722 060304          ADD      R3,R4    ;MULTIPLY NUMBER OF INNER
2577 013724 005505          ADC      R5    ; LOOPS PER OUTER LOOP BY
2578 013726 000773          BR       14$    ;NUMBER OF OUTER LOOPS PERFORMED.
2579
2580      ;*
2581      ; DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER LTC TICK.
2582 013730 016701 166370    16$:     MOV      MSTICK,R1    ;# OF MS PER LTC TICK IS DIVISOR.
2583 013734 010403          MOV      R4,R3    ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2584 013736 010502          MOV      R5,R2    ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2585 013740 004767 003336    JSR      PC,UNSDIV    ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
2586 013744 103402          BCS      18$    ;BYPASS OOPS IF WE'RE OK.
2587 013746 004767 001106    JSR      PC,OOPS    ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2588 013752 010167 166350    18$:     MOV      R1,MSLCNT    ;SET NEW VALUE FOR MS LOOP COUNT.
2589 013756 005167 000010    COM      62$    ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2590 013762 001277          BNE      2$    ;BRANCH IF ONLY ONE ITERATION DONE.
2591 013764 000261          SEC            ;SET THE SUCCESS FLAG FOR EXIT.
2592
2593 013766          60$:     PASS          ;RESTORE GPRS.
2594 013766 004736          RTS      PC          JSR      PC,0(SP)    ;RETURN TO PREG05 SUBRT.
2595 013770 000207          ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
2596 013772 000000          62$:     .WORD 0    ;2ND CALIBRATION ITERATION FLAGS.
2597 013774 000000          64$:     .WORD 0    ;DUMMY WORD FOR STORAGE OF THE READ WORD.

```

H6

CZDHV80 DHU-11 FUNC TST PART2
GLOBAL SUBROUTINEMACRO M1200 15-MAR-84 09:28 PAGE 55
- CHKBMP -

SEQ 72

```

2599 .SBTTL GLOBAL SUBROUTINE - CHKBMP -
2600 ;* *****
2601 ;* - CHECK IF CHARACTER IS A BMP CODE -
2602 ;* THIS SUBROUTINE IS USED TO CHECK FOR BMP CODES.
2603 ;* IF A BMP CODE IS DETECTED, IT WILL BE SAVED ON THE QUEUE TO BE REPORTED
2604 ;* LATER. THE CARRY IS USED AS A FLAG TO INDICATE A CODE HAS BEEN FOUND.
2605 ;*
2606 ;* INPUTS: R2 - CONTAINS THE DATA TO BE CHECKED.
2607 ;*
2608 ;* OUTPUTS: R1 - CONTAINS THE MESSAGE TO BE REPORTED.
2609 ;* ERRBLK - CONTAINS THE ERROR REPORTING ROUTINE.
2610 ;* CARRY BIT IS USED TO INDICATE A BMP CODE FOUND, CARRY SET.
2611 ;*
2612 ;* CALLING SEQUENCE: JSR PC,CHKBMP
2613 ;*
2614 ;* COMMENTS:
2615 ;*
2616 ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
2617 ;* *****
2618
2619 CHKBMP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2620 013776 004567 170056 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2621 014002 012700 170301 MOV #170301,R0 ;SET UP THE FLAGS OF A BMP CODE.
2622 014006 040200 BIC R2,R0 ;TRY TO CLEAR THE BMP CODE FLAGS.
2623 014010 001011 BNE 2$ ;IF NOT A BMP CODE, EXIT WITH FAILURE.
2624 014012 004767 002462 JSR PC,SAVBMP ;SAVE THE BMP CODE ON THE QUEUE.
2625 014016 012701 006401 MOV #EM5303,R1 ;PASS THE MESSAGE TO BE REPORTED.
2626 014022 012767 012020 170026 MOV #ER1603,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
2627 014030 000261 SEC ;PASS FLAG TO INDICATE SUCCESS, BMP CODE FOUND.
2628 014032 000401 BR 60$ ;EXIT.
2629 014034 000241 2$: CLC ;PASS FLAG TO INDICATE FAILURE.
2630 014036 010166 000004 60$: PASS R1 ;RESTORE GPRS, EXCEPT
2631 014042 004736 MOV R1,R1SLOT(SP) ;PUT R1 IN STACK SLOT.
2632 014044 000207 JSR PC,@(SP) ;RETURN TO PREG05 SUBRT.
;R1 - CONTAINS THE ADDRESS OF ERROR MESSAGE.
;CARRY BIT - SET INDICATES SUCCESS.
RTS PC

```

```

2634      SBTTL GLOBAL SUBROUTINE - CKRXTM -
2635      ;* *****
2636      ;* - CHECK RX-INT DELAY TIME -
2637      ;* THIS SUBROUTINE IS USED IN THE RXTIMER TEST AND IT CHECKS THAT THE
2638      ;* RX-INT WAS DELAYED BY +/- 20% OF THE VALUE SET IN THE RXTIMER REG.
2639      ;* IF THE ACTUAL DELAY TIME IS NOT WITHIN THIS MARGIN THEN AN ERROR
2640      ;* IS REPORTED.
2641      ;*
2642      ;* INPUTS:      R1 - REMAINING NUMBER OF MILLI SECS OF THE TIME-OUT VALUE
2643      ;*                THE TIME-OUT VALUE BEING 4*(RXTIMER VALUE + 1).
2644      ;*                R2 - RXTIMER VALUE.
2645      ;*                ERRNBR - SET TO ERROR NUMBER OF FIRST ERROR IN THIS ROUTINE.
2646      ;*                ERRBLK - SET UP BY THIS ROUTINE.
2647      ;*                EXOERR - "EXIT ON ERROR" FLAG.
2648      ;*
2649      ;* OUTPUTS:     ERRBLK - MAY BE ALTERED.
2650      ;*                EXOERR - "EXIT ON ERROR" FLAG SET IF AN ERROR DETECTED AND
2651      ;*                EXTENDED ERROR REPORTING NOT REQUESTED.
2652      ;*
2653      ;* CALLING SEQUENCE: JSR PC,CKRXTM
2654      ;*
2655      ;* COMMENTS:    THIS ROUTINE REPORTS ONE ERROR WITH THE NUMBER IN ERRNBR.
2656      ;*
2657      ;* SUBORDINATE ROUTINES USED: ER5901,UNSDIV.
2658      ;* *****
2659
2660      CKRXTM:; SAVE
2661      014046 004567 170006      JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2662      014052 010204      MOV R2,R4 ;COPY THE RXTIMER VALUE.
2663      014054 005204      INC R4 ;CALCULATE THE TIME-OUT VALUE,
2664      014056 006304      ASL R4 ; AS 4*(RXTIMER VALUE+1)
2665      014060 006304      ASL R4
2666      014062 160104      SUB R1,R4 ;CALCULATE THE NUMBER OF MS THE RX-INT WAS DELAYED.
2667
2668      ;* CALCULATE 20% OF THE RXTIMER VALUE.
2669      ;*
2670      014064 012701 000005      MOV #5,R1 ;PASS THE DIVISOR TO THE SUBR.
2671      014070 010203      MOV R2,R3 ;LOAD THE RXTIMER VALUE INTO THE LSW OF THE DIVIDEND.
2672      014072 005002      CLR R2 ;CLEAR THE MSW OF THE DIVIDEND.
2673      014074 004767 003202      JSR PC,UNSDIV ;DIVIDE THE RXTIMER VALUE BY 5.
2674      014100 010302      MOV R3,R2 ;RESTORE THE RXTIMER VALUE.
2675      014102 010305      MOV R3,R5 ;COPY THE RXTIMER VALUE.
2676      014104 160103      SUB R1,R3 ;REDUCE IT BY 20% TO OBTAIN THE LOLIMIT OF THE
2677      014106 060105      ADD R1,R5 ;PERMISSIBLE TIME DELAY.
2678      014110 062705 000002      ADD #2,R5 ;INCREASE THE RXTIMER VALUE BY 20% AND ADD A
2679      ;* ;FURTHER 2 MS TO OBTAIN THE HILIMIT OF THE
2680      ;* ;PERMISSIBLE TIME DELAY.
2681      ;*
2682      ;* CHECK IF THE RX-INT OCCURED WITHIN THE CALCULATED PERMISSIBLE LIMITS.
2683      ;*
2684      014114 020504      CMP R5,R4 ;COMPARE HILIM WITH ACTUAL RX-INT DELAY.
2685      014116 002402      BLT #2 ;REPORT THE ERROR IF THE DELAY WAS TOO LONG.
2686      014120 020304      CMP R3,R4 ;COMPARE LOLIM WITH ACTUAL RX-INT DELAY.
2687      014122 002417      BLT #60 ;AVOID ERROR IF DELAY WAS WITHIN THE LIMITS.
2688      ;*
2689      ;* REPORT THE ERROR, INCORRECT DELAY ON RX-INT.
2690      ;*

```

J6

CZDHVBO DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 56-1
GLOBAL SUBROUTINE - CKRXTM -

SEQ 74

```

2690 014124 010201          2$:  MOV    R2,R1      ;PASS THE EXPECTED VALUE OF THE RX-INT DELAY.
2691 014126 010402          MOV    R4,R2      ;PASS THE ACTUAL VALUE OF THE RX-INT DELAY.
2692 014130 012703 007003    MOV    $EM5802,R3  ;PASS THE MESSAGE,
2693                                ; "RXTIMER BAD, INCORRECT MILLI SEC DELAY ON
2694                                ; RX-INT".
2695 014134 012767 012170 167714  MOV    $ER5901,ERRBLK ;SET UP THE ERROR REPORTING ROUTINE.
2696 014142                                TRAP    C$ERROR
2697 014142 104460
2698 014144 032767 000100 166036  BIT    $BIT06,OPTION ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
2699 014152 001003          BNE    60$        ;AVOID SETTING THE FLAG IF IT HAS.
2700 014154 012767 000001 166070  MOV    $1,EXDERR  ;SET THE "EXIT ON ERROR" FLAG.
2701 014162          60$:  PASS
2702 014164 004736          JSR    PC,$(SP)+    ;RETURN TO PREG05 SUBRT.
2702 014164 000207          RTS    PC          ;RETURN FROM SUBROUTINE.

```

K6

CZDHV80 DHU-11 FUNC TST PART2
GLOBAL SUBROUTINEMACRO M1200 15-MAR-84 09:28 PAGE 57
- CKTRAP -

SEQ 75

```

2704 .SBTTL GLOBAL SUBROUTINE - CKTRAP -
2705 ;*****
2706 ;* CHECK TRAP ROUTINE -
2707 ;* THIS SUBROUTINE IS USED TO CHECK FOR A BUS TIME-OUT TRAP (004 TRAP)
2708 ;* WHICH IS CAUSED BY AN ACCESS TO A NON-EXISTENT MEMORY OR I/O LOCATION.
2709 ;* IF THE TRAP DOES NOT OCCUR, THIS ROUTINE RETURNS A SUCCESS INDICATION.
2710 ;*
2711 ;* INPUTS: R0 - SOURCE ADDRESS FOR MOVE.
2712 ;* R1 - DESTINATION ADDRESS FOR MOVE.
2713 ;* (R0) - SOURCE FOR THE MOVE.
2714 ;*
2715 ;* OUTPUTS: (R1) - WRITTEN TO THE CONTENTS OF (R0).
2716 ;* CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED.
2717 ;* TP4FLG - NONZERO IF TRAP OCCURRED, CLEARED OTHERWISE.
2718 ;*
2719 ;* CALLING SEQUENCE: JSR PC,CKTRAP
2720 ;*
2721 ;* COMMENTS: IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS WHICH
2722 ;* IS LABELED ADRPTR WILL BE THE TRAP PC ADDRESS ON THE STACK.
2723 ;*
2724 ;* SUBORDINATE ROUTINES CALLED: NONE.
2725 ;*****
2726
2727 014166 CKTRAP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
014166 004567 167666 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2728 014172 005067 166102 CLR TP4FLG ;CLEAR THE 004 TRAP FLAGS.
2729 014176 011011 MOV (R0),(R1) ;PERFORM THE MOVE IN QUESTION.
2730 014200 005767 166074 ADRPTR:: TST TP4FLG ;CHECK FOR OCCURENCE OF TRAP.
2731 014204 000261 SEC ;INDICATE SUCCESS.
2732 014206 001401 BEQ 60$ ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
2733 014210 000241 CLC ;INDICATE FAILURE.
2734 014212 60$: PASS ;RESTORE GPRS.
014212 004736 JSR PC,8(SP)+ ;RETURN TO PREG05 SUBRT.
2735 014214 000207 RTS PC

```

```

2737 .SBTTL GLOBAL SUBROUTINE - CLNRST -
2738 ;*****
2739 ;* - CLEAN RESET OF THE DEVICE UNDER TEST -
2740 ;* THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
2741 ;* THE DUT'S SELF-TEST IS SKIPPED, AND THE FIFO IS PURGED OF ANY ERROR
2742 ;* CODES, ETC.
2743 ;* IF THE RESET DOES NOT SUCCESSFULLY COMPLETE, THEN THE CARRY BIT IS
2744 ;* PASSED BACK TO THE CALLING ROUTINE (CLEAR).
2745 ;*
2746 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR
2747 ;* TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
2748 ;* ERRNBR - ERROR NUMBER FOR POSSIBLE ERROR REPORT.
2749 ;* ERRBLK - ERRTP,ERNBR,AND ERRMSG SET UP CORRECTLY.
2750 ;*
2751 ;* OUTPUTS: THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
2752 ;* CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
2753 ;* ERRBLK - VALUE MAY BE DESTROYED.
2754 ;* IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
2755 ;* TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
2756 ;*
2757 ;* CALLING SEQUENCE: JSR PC,CLNRST
2758 ;*
2759 ;* COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS ERRNBR.
2760 ;* THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
2761 ;*
2762 ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET,PUFIFO,RESETT.
2763 ;*****
2764
2765 014216 CLNRST:: SAVE JSR R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
014216 004567 167636 ;CALL REGISTER SAVE SUBRT.
2766
2767 ; RESET THE DUT.
2768 ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS FROM ERRNBR THRU ERRNBR+2
2769 ;
2770 014222 004767 002054 JSR PC,RESETT ;RESET THE DUT TO A KNOWN STATE.
2771 014226 103002 BCC 60$ ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
2772
2773 ; PURGE THE FIFO OF ERROR CODES, SAVE ANY BMP CODES FOUND.
2774 ;
2775 014230 004767 001134 JSR PC,PUFIFO ;PURGE THE FIFO.
2776
2777 014234 60$: ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
2778 014234 PASS ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
014234 004736 JSR PC,0(SP). ;RETURN TO PREG05 SUBRT.
2779 ;CARRY BIT;IF CLEAR, THEN ABORT THE TEST.
2780 014236 C00207 RTS PC

```

```

2782 .SBTTL GLOBAL SUBROUTINE - CLR16W -
2783 ; * *****
2784 ; * - CLEAR SIXTEEN WORDS ROUTINE -
2785 ; * THIS SUBROUTINE CLEARS 16 WORDS STARTING WITH THE SPECIFIED WORD.
2786 ; *
2787 ; * INPUTS: R0 - ADDRESS OF THE FIRST WORD TO CLEAR.
2788 ; *
2789 ; * OUTPUTS: (R0) TO (R0+15) - 16 WORDS OF MEMORY ARE CLEARED TO 0.
2790 ; *
2791 ; * CALLING SEQUENCE: JSR PC,CLR16W
2792 ; *
2793 ; * COMMENTS:
2794 ; *
2795 ; * SUBORDINATE ROUTINES CALLED: NONE.
2796 ; * - *****
2797
2798 014240 CLR16W:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
014240 004567 167614 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2799 014244 012701 000020 ;SET THE LOOP COUNTER TO 16.
2800 014250 005020 2$: MOV #16.,R1 ;CLEAR A WORD OF MEMORY.
2801 014252 005301 CLR (R0)+ ;COUNT THIS LOOP.
2802 014254 001775 DEC R1 ;LOOP IF NOT 16 WORD CLEARED.
2803 014256 BNE 2$ ;RESTORE GPRS.
014256 004736 60$: PASS JSR PC,8(SP)+ ;RETURN TO PREG05 SUBRT.
2804 014260 000207 RTS PC

```

```

2806      ,SBTTL  GLOBAL SUBROUTINE      - DELAY -
2807      ;*****
2808      ;*                                     - DELAY SUBROUTINE -
2809      ;*      THIS SUBROUTINE IS USED TO DELAY A VARIABLE NUMBER OF MILLI-SECONDS.
2810      ;*
2811      ;*  INPUTS:      R4 - CONTAINS THE NUMBER OF MS TO DELAY.
2812      ;*                  MSLCNT.
2813      ;*
2814      ;*  OUTPUTS:     NONE.
2815      ;*
2816      ;*  CALLING SEQUENCE:  JSR      PC,DELAY
2817      ;*
2818      ;*  COMMENTS:     IF NO HARDWARE CLOCK INTERRUPTS ARE OCCURRING, CONTROL-CS WILL
2819      ;*                  NOT BE HONORED FOR THE DURATION OF THE DELAY.
2820      ;*
2821      ;*  SUBORDINATE ROUTINES CALLED: NONE.
2822      ;*****
2823
2824      014262      DELAY:: SAVE                                ;SAVE CONTENTS OF GPRS R0 THRU R5.
2825      014262      004567 167572                                R5,PREG05      ;CALL REGISTER SAVE SUBRT.
2826      014266      010401      MOV      R4,R1                ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
2827      014270      012702 177777      MOV      0-1,R2        ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
2828      014274      005003      CLR      R3                    ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
2829      014276      012704 014320      MOV      062$,R4        ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
2830      014302      004767 000536      JSR      PC,MSLOOP      ;DELAY THE REQUESTED 0 OF MS.
2831      014310      004767 000544      BCC      60$             ;EXIT ROUTINE IF WE TIMED-OUT.]
2832      014314      004736      JSR      PC,00PS               ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
2833      014316      000207      60$: PASS                      ;RESTORE GPRS.
2834      014316      000207      RTS      PC                    JSR      PC,0(SP)+      ;RETURN TO PREG05 SUBRT.
2835      014320      177777      62$: .WORD      -1            ;DUMMY, NON-ZERO WORD.

```

```

2837 .SBTTL GLOBAL SUBROUTINE - DMABUF -
2838 *** *****
2839 |* - DMA FROM ADDR BUFBAS -
2840 |* THIS SUBROUTINE INITIATES A DMA FROM THE ADDRESS BUFBAS AND WAITS
2841 |* FOR IT TO COMPLETE AND THE LAST CHARACTER TO BE RECIEVED. THE LINE
2842 |* ON WHICH TO PERFORM THE DMA AND THE NUMBER OF CHARACTERS TO TRANSMIT
2843 |* ARE PASSED AS INPUT PARAMETERS.
2844 |*
2845 |* INPUTS: R1 - LINE NUMBER ON WHICH TO PERFORM THE DMA.
2846 |* R3 - NUMBER OF CHARACTERS TO TRANSMIT.
2847 |* CSRA - CONTAINS THE ADDRESS OF THE CSR.
2848 |* ERRNBR - CONTAINS THE ERROR NUMBER OF THE FIRST ERROR
2849 |* IN THIS SUBROUTINE.
2850 |*
2851 |* OUTPUTS: CARRY - SUCCESS FLAG - SET IF DMA COMPLETED SUCCESSFULLY.
2852 |* CLEAR IF DMA-START BIT FOUND SET OR
2853 |* NO TX-ACTION RECIEVED.
2854 |* ERRNBR - CONTENTS ALTERED TO INITIAL ERRNBR + 1.
2855 |*
2856 |* CALLING SEQUENCE: JSR PC,DMABUF
2857 |*
2858 |* COMMENTS: THIS ROUTINE RETURNS CARRY CLEAR IF AN ERROR OCCURED, WITH
2859 |* THE APPROPRIATE ERROR NUMBER IN ERRNBR. A NON-RELATED TEST
2860 |* ERROR SHOULD BE REPORTED, UPON RETURN TO THE TEST, IF THE CARRY
2861 |* FLAG IS CLEAR, THIS ROUTINE REPORTS ERRORS,
2862 |* INITIAL ERRNBR THRU INITIAL ERRNBR + 1.
2863 |*
2864 |* SUBORDINATE ROUTINES CALLED: DELAY,DOOMA,WAIBIS.
2865 |* *****
2866
2867 014322 DMABUF:: SAVE
2868 014322 004567 167532 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2869 014326 012702 002650 MOV #BUFBAS,R2 ;PASS THE START ADDR OF THE BUFFER TO TX.
2870 014332 004767 000042 JSR PC,DOOMA ;INITIATE THE DMA.
2871 014336 103016 BCC 601 ;RETURN WITH FAILURE IF DMA START BIT FOUND SET.
2872
2873 |* WAIT FOR THE DMA TO COMPLEATE I.E A TX-ACTION BEING RETURNED, AND THEN WAIT
2874 |* FOR THE LAST CHARACTER TO BE RECIEVED.
2875
2876 014340 005267 167506 INC ERRNBR ;SET THE ERROR NUMBER TO INITIAL ERRNBR + 1.
2877 014344 012701 170536 MOV #170536,R1 ;INDICATE TO TEST BIT15 WITH TIME OUT OF 350 MS
2878 014350 016702 165652 MOV CSRA,R2 ;PASS THE ADDR OF THE REG TO TEST.
2879 014354 004767 003056 JSR PC,WAIBIS ;WAIT FOR THE TX-ACTION.
2880 014360 103025 BCC 601 ;RETURN WITH FAILURE IF NO TX-ACTION FOUND.
2881 014362 012704 000005 MOV #5,R4 ;SET THE DELAY OF 5 MS.
2882 014366 004767 177670 JSR PC,DELAY ;WAIT FOR THE LAST CHARACTER TO BE RECIEVED.
2883 014374 000261 SEC ;SET THE CARRY FLAG TO INDICATE SUCCESS.
2884 014376 000207 601: PASS
2885 014374 004736 JSR PC,@(SP), ;RETURN TO PREG05 SUBRT.
2886 014376 000207 RTS PC ;RETURN.

```

C7

C7DMV80.DMU.11.FUNC.TST.PART2 MACRO.M1200 15-MAR-84 09:28 PAGE 62

SEQ 80

GLOBAL SUBROUTINE

```

2886 .SBTTL GLOBAL SUBROUTINE - DODMA -
2887 * *****
2888 * - INITIATE DMA TRANSMISSION ROUTINE -
2889 * THIS ROUTINE WRITES THE DMA PARAMETER TO THE SPECIFIED DEVICE AND
2890 * INITIATES THE DMA TRANSMISSION.
2891 *
2892 * INPUTS: R1 - LINE NUMBER ON WHICH TO INITIATE THE DMA.
2893 * R2 - START ADDRESS OF THE DMA BUFFER (16 BIT VIRTUAL).
2894 * R3 - CHARACTER COUNT OF THE DMA BUFFER.
2895 * CSRA - CONTAINS ADDRESS OF THE DUT CSR.
2896 * IESTAT - STORAGE FOR STATES OF THE INTERRUPT ENABLE BITS.
2897 * TXAD1A - CONTAINS ADDRESS OF DMA TX BUFFER ADDRESS REG #1.
2898 * TXAD2A - CONTAINS ADDRESS OF DMA TX BUFFER ADDRESS REG #2.
2899 * TXBFCA - CONTAINS ADDRESS OF DMA CHARACTER COUNT REGISTER.
2900 *
2901 * OUTPUTS: CARRY - SUCCESS FLAG (SET IF DMA_START FOUND CLEAR).
2902 * DUT TBUFFAD1 - LS 16 BITS OF DMA BUFFER ADDRESS (INITIALIZED).
2903 * DUT TBUFFAD2 - MS 6 BITS OF DMA BUFFER ADDRESS (INITIALIZED),
2904 * DMA_START BIT SET.
2905 * DUT TBUFFCT - DMA BUFFER CHARACTER COUNT (INITIALIZED).
2906 *
2907 * CALLING SEQUENCE: JSR PC,DODMA
2908 *
2909 * COMMENTS: THIS ROUTINE ASSUMES MEMORY MANAGEMENT IS DISABLED AND
2910 * CLEARS THE TWO MSB OF THE DMA ADDRESS, I.E. BITS 0 AND 1
2911 * OF THE TBUFFAD2 REG.
2912 *
2913 * SUBORDINATE ROUTINES CALLED: NONE.
2914 * - - - - -
2915
2916 014400 DODMA:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2917 014400 004567 167454 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2918 014404 012704 000200 MOV #200,R4 ;PREPARE TO CLEAR UPPER 6 BITS OF DMA BUFF ADR.
2919
2920 * WRITE THE DMA PARAMETERS OUT TO THE DUT DMA REGISTERS.
2921 * DISABLE INTERRUPTS.
2922 * SET UP DUT CSR IND.ADR.REG FIELD.
2923 * WRITE THE DMA TRANSMIT CHARACTER COUNT.
2924 * WRITE THE LEAST SIGNIFICANT 16 BITS OF THE DMA BUFFER START ADDRESS.
2925 * WRITE THE MOST SIGNIFICANT 6 BITS OF THE ADDRESS.
2926 * SETTING THE DMA_START BIT, AND INITIATING THE DMA TRANSMISSION.
2927
2928 014410 64: GETPRI R5 ;GET THE PRESENT PROCESSOR PRIORITY.
2929 014410 104440 TRAP C:GPRI
2930 014412 010005 MOV R0,R5
2931 014414 SETPRI #PRI07 ;DISABLE ALL HARDWARE INTERRUPTS.
2932 014414 012700 00034C MOV #PRI07,R0
2933 014420 104441 TRAP C:SPRI
2934 014422 056701 165630 BIS IESTAT,R1 ;PREPARE FOR SETUP OF LINE NUMBER IN DUT CSR.
2935 014426 010177 165574 MOV R1,BCSRA ;SET UP THE DUT CSR IND.ADR.REG FIELD.
2936 014432 105777 165604 MOV R3,GTXB2A ;TEST THE DUT DMA_START BIT.
2937 014436 000241 ;INDICATE FAILURE IN CASE DMA.MO BIT IS SET.
2938 014440 100411 ;EXIT WITH FAILURE IF DMA.MO BIT IS SET.
2939 014442 010377 165576 MOV R3,GTXB2A ;WRITE THE DMA CHARACTER COUNT.
2940 014446 010277 165566 MOV R2,GTXB1A ;WRITE THE LS 16 BITS OF BUFFER ADDRESS.
2941 014452 110477 165564 MOV R4,GTXB2A ;WRITE MS 6 BITS OF ADR AND START DMA TX.

```

07

2938	014456		SETPRI	R5		,RESTORE THE PROCESSOR PRIORITY.		
	014456	010500					MOV	R5,R0
	014460	104441					TRAP	C\$SPRI
2939	014462	000261	SEC			,INDICATE SUCCESS.		
2940								
2941	014464		60\$: PASS		JSR	,RESTORE GPRS,		
	014464	004736				PC,5(SP),		,RETURN TO PREG05 SUBRT.
2942	014466	000207	RTS	PC		, CARRY - SUCCESS FLAG (SET IF SUCCESS),		

E7

CZDHV80 DHU-11 FUNC TST PART2
GLOBAL SUBROUTINEMACRO M1200 15-MAR-84 09:28 PAGE 63
- FINACT -

SEQ 82

```

2944 .SBTTL GLOBAL SUBROUTINE - FINACT -
2945 ;* *****
2946 ;* - FIND FIRST ACTIVE LINE -
2947 ;* THIS SUBROUTINE CALCULATES THE NUMBER OF THE FIRST ACTIVE LINE THAT
2948 ;* IS FOUND IN THE ACTIVE LINE BIT MAP ACTLNS.
2949 ;*
2950 ;* INPUTS: ACTLNS - CONTAINS THE ACTIVE LINE BIT MAP.
2951 ;*
2952 ;* OUTPUTS: R1 - CONTAINS THE NUMBER OF THE FIRST ACTIVE LINE.
2953 ;* R5 - CONTAINS THE BIT MAP REPRESENTATION OF THE ACTIVE LINE.
2954 ;* CARRY SET INDICATES SUCCESS.
2955 ;*
2956 ;* CALLING SEQUENCE: JSR PC,FINACT
2957 ;*
2958 ;* COMMENTS:
2959 ;*
2960 ;* SUBORDINATE ROUTINES CALLED: NONE.
2961 ;* - *****
2962
2963 014470 FINACT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
014470 004567 167364 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2964
2965 ;* FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
2966 ;*
2967 014474 005001 CLR R1 ;CLEAR THE LINE NUMBER COUNTER.
2968 014476 012703 000020 MOV #NUMLNS,R3 ;GET MAX LINE NUMBER.
2969 014502 016700 165512 MOV ACTLNS,R0 ;GET THE ACTIVE LINE BIT MAP.
2970 014506 012705 000001 MOV #1,R5 ;SET UP A LINE BIT MASK.
2971 014512 030500 2$: BIT R5,R0 ;LOOK FOR AN ACTIVE LINE.
2972 014514 001006 BNE 4$ ;BRANCH TO BEGIN TEST IF A LINE HAS BEEN FOUND.
2973 014516 006305 ASL R5 ;SHIFT THE BIT MASK FOR THE NEXT LINE.
2974 014520 005201 INC R1 ;INCREMENT THE LINE NUMBER COUNTER.
2975 014522 020103 CMP R1,R3 ;CHECK IF ALL LINES HAVE BEEN TRIED.
2976 014524 002772 BLT 2$ ;LOOP TO TRY THE NEXT LINE.
2977 014526 000241 CLC ;CLEAR CARRY BIT, NO ACTIVE LINE FOUND.
2978 014530 000401 BR 60$ ;EXIT WITH FAILURE.
2979 014532 000261 4$: SEC ;SET CARRY, SUCCESS.
2980
2981 014534 60$: PASS R1,R5 ;RESTORE GPRS, EXCEPT
014534 010166 000004 MOV R1,R1SLOT(SP) ;PUT R1 IN STACK SLOT.
014540 010566 000014 MOV R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
014544 004736 JSR PC,0(SP) ;RETURN TO PREG05 SUBRT.
2982 ;R1 - CONTAINS THE NUMBER OF FIRST ACTIVE LINE.
2983 ;R5 - CONTAINS THE BIT MAP OF THE ACTIVE LINE.
2984 ;CARRY - SET INDICATES SUCCESS.
2985 014546 000207 RTS PC

```

F7

CZDHV80 DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 64
GLOBAL SUBROUTINE - INDATP -

SEQ 83

```

2987 .SBTTL GLOBAL SUBROUTINE - INDATP -
2988 ;** *****
2989 ;* - INITIALISE DATA PATTERN -
2990 ;* THIS SUBROUTINE IS USED TO INITIALISE AN INCREMENTAL BYTE DATA PATTERN
2991 ;* IN THE GENERAL BUFFER AREA.
2992 ;* THE DATA PATTERN WILL BE SEQUENTIAL FROM 0 TO 255 (DECIMAL).
2993 ;*
2994 ;* INPUTS: BUFBA5 - ADDRESS OF THE START OF THE GENERAL BUFFER AREA.
2995 ;*          BUFMD - ADDRESS OF THE 255 TH LOCATION.
2996 ;*
2997 ;* OUTPUTS: THE FIRST 255 LOCATIONS OF THE GENERAL BUFFER AREA CONTAIN DATA
2998 ;*
2999 ;* CALLING SEQUENCE: JSR PC,INDATP
3000 ;*
3001 ;* COMMENTS:
3002 ;*
3003 ;* SUBORDINATE ROUTINES CALLED: NONE.
3004 ;-- *****
3005
3006 014550 INDATP:; SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
014550 004567 167304 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3007
3008 014554 012702 002650 MOV #BUFBA5,R2 ;INITIALIZE THE DATA PATTERN IN THE GENERAL
3009 014560 005003 CLR R3 ; DATA BUFFER TO A 256 BYTE PATTERN.
3010 014562 110322 2$: MOVB R3,(R2)+ ;
3011 014564 005203 INC R3 ;SELECT THE NEXT CHARACTER.
3012 014566 020227 003250 CMP R2,#BUFMD ;CHECK IF WE HAVE 256 DATA PATTERNS.
3013 014572 103773 BLO 2$ ;
3014
3015 014574 60$: PASS ;RESTORE GPRS.
014574 004736 JSR PC,#(SP)+ ;RETURN TO PREG05 SUBRT.
3016 014576 000207 RTS PC

```

67

CZDHV80 DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 65
GLOBAL SUBROUTINE - INDPX -

SEQ 84

```

3018 .SBTTL GLOBAL SUBROUTINE - INDPX -
3019 ;* *****
3020 ;* - INITIALISE DATA PATTERN WITHOUT XON OR XOFF
3021 ;* THIS SUBROUTINE IS USED TO INITIALISE AN INCREMENTAL BYTE DATA PATTERN
3022 ;* IN THE GENERAL BUFFER AREA.
3023 ;* THE DATA PATTERN WILL BE FROM 0 TO 255, BUT WILL EXCLUDE THE FOLLOWING
3024 ;* TWO CHARACTERS, (ASCII DC1, DC3) XON AND XOFF. THIS WILL CAUSE THE
3025 ;* LAST TWO DATA CHARACTERS TO BE THE SAME AS THE FIRST TWO.
3026 ;*
3027 ;* INPUTS: BUFBAS - ADDRESS OF THE START OF THE GENERAL BUFFER AREA.
3028 ;*          BUF MID - ADDRESS OF THE 255 TH LOCATION.
3029 ;*
3030 ;* OUTPUTS: THE FIRST 255 LOCATIONS OF THE GENERAL BUFFER AREA CONTAIN DATA
3031 ;*
3032 ;* CALLING SEQUENCE: JSR PC,INDPX
3033 ;*
3034 ;* COMMENTS:
3035 ;*
3036 ;* SUBORDINATE ROUTINES CALLED: NONE.
3037 ;* - *****
3038
3039 014600 004567 167254 INDPX:: SAVE JSR R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
3040 ;* ;CALL REGISTER SAVE SUBRT.
3041 ;*
3042 ;* INITIALIZE THE 256 BYTE DATA PATTERN.
3043 ;* ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
3044 ;* NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
3045 ;* -
3045 014604 012702 002650 MOV #0UFBAS,R2 ;INITIALIZE THE DATA PATTERN IN THE GENERAL
3046 014610 005003 CLR R3 ; DATA BUFFER TO A 256 BYTE PATTERN.
3047 014612 110322 2$: MOV R3,(R2)+ ;
3048 014614 105203 INCB R3 ;SELECT THE NEXT CHARACTER.
3049 014616 122703 000021 CMPB #21,R3 ;CHECK FOR AN XON CHARACTER.
3050 014622 001001 BNE 4$ ;BRANCH IF CHAR NOT AN XON.
3051 014624 105203 INCB R3 ;FORCE THE NEXT CHARACTER.
3052 014626 122703 000023 4$: CMPB #23,R3 ;CHECK FOR AN XOFF CHARACTER.
3053 014632 001001 BNE 6$ ;BRANCH IF NOT AN XOFF CHARACTER.
3054 014634 105203 INCB R3 ;FORCE THE NEXT CHARACTER.
3055 014636 020227 003250 6$: CMP R2,#BUFMID ;CHECK IF WE HAVE 256 DATA PATTERNS.
3056 014642 103763 BLO 2$ ;
3057
3058 014644 60$: PASS ;RESTORE GPRS.
3059 014644 004736 JSR PC,#(SP)+ ;RETURN TO PREG05 START.
3059 014646 000207 RTS PC

```

```

3061 .SBTTL GLOBAL SUBROUTINE - LINBIT -
3062 ;* *****
3063 ;* - LINE NUMBER TO BIT MAP CONVERSION SUBROUTINE -
3064 ;* THIS SUBROUTINE IS USED TO GENERATE A BIT MAP (ONE BIT OF 16 SET)
3065 ;* BASED ON A LINE NUMBER (RANGE: 1 TO 16). ONLY THE LS 4 BITS OF THE
3066 ;* LINE NUMBER WORD ARE USED, THE OTHERS ARE MASKED OUT (SO UNMASKED
3067 ;* MSBYTES OF OUT CSRS CAN BE PASSED TO THIS ROUTINE WITHOUT ERROR).
3068 ;*
3069 ;* INPUTS: R1 - LINE NUMBER (ONLY LS 4 BITS USED, OTHERS DISREGARDED).
3070 ;* BITTBL - BASE LABEL OF A 16 WORD BIT TABLE.
3071 ;*
3072 ;* OUTPUTS: R0 - BIT MAP, BIT CORRESPONDING TO LINE NUMBER IS SET:
3073 ;* IF LINE NUMBER IS 3, THEN BIT3 IS SET, ETC.
3074 ;*
3075 ;* CALLING SEQUENCE: JSR PC,LINBIT
3076 ;*
3077 ;* COMMENTS: NO CHECKING IS PERFORMED TO VERIFY THAT THE LINE NUMBER IS
3078 ;* A LEGAL LINE NUMBER FOR THE OUT (IE - LESS THAN NUMLNS).
3079 ;* NOTE: THE LINE NUMBER IS NOT DESTROYED OR ALTERED, SO THIS
3080 ;* ROUTINE CAN BE USED EASILY IN LOOPS.
3081 ;*
3082 ;* SUBORDINATE ROUTINES CALLED: NONE.
3083 ;* *****
3084
3085 014650 LINBIT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3086 014650 004567 167204 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3087 014654 042701 177760 BIC #177760,R1 ;MASK OUT ALL BUT 4 LSBITS OF THE LINE #.
3088 014660 006301 ASL R1 ;MULTIPLY LINE # BY 2 TO GET WORD TABLE OFFSET.
3089 014662 016100 002336 MOV BITTBL(R1),R0 ;GET THE SINGLE BIT BIT MAP.
3090 014666 010066 000002 60$: PASS R0 ;RESTORE GPRS, EXCEPT THE FOLLOWING.
3090 014672 004736 MOV R0,R0SLOT(SP) ;PUT R0 IN STACK SLOT.
3090 014674 000207 RTS PC,0(SP) ;RETURN TO PREG05 SUBRT.
;R0 - BIT MAP WITH LINE # BIT SET.

```

CZDHVBO DHU-11 FUNC TST PART2
GLOBAL SUBROUTINE

MACRO M1200 15-MAR-84 09:28 PAGE 67
- MAPCNT -

SEQ 86

```

3092 .SBTTL GLOBAL SUBROUTINE - MAPCNT -
3093 ; * *****
3094 ; * - COUNT BITS IN BIT MAP ROUTINE -
3095 ; * THIS SUBROUTINE COUNTS THE NUMBER OF BITS WHICH ARE SET IN A BIT MAP.
3096 ; *
3097 ; * INPUTS: R2 - THE BIT MAP FOR WHICH TO COUNT THE BITS.
3098 ; *
3099 ; * OUTPUTS: R2 - COUNT OF THE NUMBER OF BITS THAT WERE SET.
3100 ; *
3101 ; * CALLING SEQUENCE: JSR PC,MAPCNT
3102 ; *
3103 ; * COMMENTS:
3104 ; *
3105 ; * SUBORDINATE ROUTINES CALLED: NONE.
3106 ; * - *****
3107
3108 014676 MAPCNT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
      014676 004567 167156 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3109 014702 010201 MOV R2,R1
3110 014704 001405 BEQ 60$ ;EXIT WITH ZERO IF NO BITS ARE SET IN MAP.
3111
3112 014706 005002 CLR R2 ;CLEAR THE BIT COUNT.
3113 014710 000261 SEC ;COUNT THE LAST BIT TO BE SHIFTED OUT.
3114
3115 014712 005502 2$: ADC R2 ;COUNT THE BIT IF IT WAS SET.
3116 014714 006301 ASL R1 ;SHIFT ANOTHER BIT OUT OF THE MAP.
3117 014716 001375 BNE 2$ ;LOOP IF ALL BITS NOT SHIFTED OUT OF MAP.
3118
3119 014720 60$: PASS R2 ;RESTORE GPRS, EXCEPT THE FOLLOWING:
      014720 010266 000006 MOV R2,R2SLOT(SP) ;PUT R2 IN STACK SLOT.
      014724 004736 JSR PC,0(SP) ;RETURN TO PREG05 SUBRT.
3120 014726 000207 RTS PC ; R2 - COUNT OF BITS SET IN BIT MAP.

```

```

3122 .SBTTL GLOBAL SUBROUTINE - MSLGET -
3123 ;*****
3124 ;* - MILLI SECONDS LOOP WHICH RETURNS READ WORD AND REMAINING TIME -
3125 ;* THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
3126 ;* TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD. THE
3127 ;* CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
3128 ;* DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
3129 ;* THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
3130 ;* ROUTINE AND THEN ONCE EACH MILLI-SECOND THERE AFTER.
3131 ;* UPON RETURN, THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION
3132 ;* IS RETURNED BY THIS SUBROUTINE.
3133 ;*
3134 ;* INPUTS: R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
3135 ;* R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
3136 ;* R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
3137 ;* R4 - ADDRESS OF THE WORD TO TEST.
3138 ;* MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
3139 ;*
3140 ;* OUTPUTS: R0 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
3141 ;* R1 - REMAINING NUMBER OF MS IN TIME-OUT TIME.
3142 ;* CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
3143 ;*
3144 ;* CALLING SEQUENCE: JSR PC,MSLGET
3145 ;*
3146 ;* COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
3147 ;* CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
3148 ;* ON THE SYSTEM.
3149 ;* THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
3150 ;* DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
3151 ;* LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
3152 ;* IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
3153 ;* THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
3154 ;* IF THE CONDITION IS MET, FAILURE OTHERWISE.
3155 ;*
3156 ;*
3157 ;* SUBORDINATE ROUTINES CALLED: NONE.
3158 ;*****
3159
3160 014730 MSLGET:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
014730 004567 167124 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3161 ;*
3162 ;* SET UP MASK FOR REMOVING UNUSED BITS IN THE TEST WORD, AND CLEAR UNUSED
3163 ;* BITS IN THE DESIRED STATE WORD TO ALLOW DIRECT COMPARISON.
3164 ;*
3165 014734 005102 COM R2 ;GET MASK OF UNUSED BITS.
3166 014736 040203 BIC R2,R3 ;MASK OUT UNUSED BITS IN DESIRED STATE WORD.
3167 ;*
3168 ;* HANDLE THE TEST AND EXIT IF WE HAVE A 0 TIME-OUT VALUE.
3169 ;*
3170 014740 005701 TST R1 ;TEST THE TIME-OUT VALUE FOR ZERO.
3171 014742 001011 BNE Z$ ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
3172 014744 011400 MOV (R4),R0 ;GET THE WORD TO TEST BEFORE EXITING.
3173 014746 010067 000070 MOV R0,R2$ ;SAVE VALUE SO WE CAN RETURN IT.
3174 014752 040200 BIC R2,R0 ;MASK OUT UNTESTED BITS OF WORD.
3175 014754 020003 CMP R0,R3 ;COMPARE AGAINST DESIRED STATE WORD.
3176 014756 000261 SEC ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
3177 014760 001420 BEQ B$ ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.

```

K7

CZDHVBO DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 68-1
GLOBAL SUBROUTINE - MSLGET

SEQ 88

```

3178 014762 000241          CLC          ;INDICATE FAILURE (TIME-OUT).
3179 014764 000416          BR          6$ ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
3180                                ;*
3181                                ; NON-ZERO TIME-OUT VALUE. LOOP, WAITING FOR CONDITION OR TIME-OUT.
3182                                ;*
3183 014766 016705 165334    2$: MOV      MSLCNT,R5 ;LOAD MS LOOP COUNT.
3184 014772 011400          4$: MOV      (R4),R0 ;GET THE WORD TO TEST.
3185 014774 010067 000042    MOV      R0,62$ ;SAVE WORD IN CASE THIS IS THE LAST.
3186 015000 040200          BIC      R2,R0 ;MASK OUT UNTESTED BITS OF WORD.
3187 015002 020003          CMP      R0,R3 ;COMPARE AGAINST DESIRED STATE WORD.
3188 015004 000261          SEC          ;SET CARRY IN CASE OF SUCCESS.
3189 015006 001405          BEQ      6$ ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
3190 015010 005305          DEC      R5 ;COUNT DOWN THE INSIDE MS LOOP COUNT.
3191 015012 001367          BNE      4$ ;LOOP IF MS NOT UP.
3192 015014 005301          DEC      R1 ;DECREMENT THE MS TIME COUNT.
3193 015016 001363          BNE      2$ ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
3194 015020 000241          CLC          ;CLEAR CARRY, WE TIMED-OUT.
3195                                ;*
3196                                ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME-OUT VALUE).
3197                                ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD. EXIT ROUTINE.
3198                                ;*
3199 015022 016700 000014    6$: MOV      62$,R0 ;PASS OUT THE LAST READ WORD.
3200 015026 010066 000002    60$: PASS    R0,R1 ;RESTORE GPRS, EXCEPT THE FOLLOWING:
                                MOV      R0,R0SLOT(SP) ;PUT R0 IN STACK SLOT.
                                MOV      R1,R1SLOT(SP) ;PUT R1 IN STACK SLOT.
                                JSR      PC,@(SP)+ ;RETURN TO PREGOS SUBRT.
3201                                ;R0 - LAST READ WORD CHECKED FOR CONDITION.
3202                                ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
3203 015040 000207          RTS      PC ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
3204                                ;*
3205                                ; LOCAL STORAGE.
3206                                ;*
3207 015042 000000          62$: .WORD    0 ;STORAGE FOR THE LAST READ WORD.

```

```

3209 .SBTTL GLOBAL SUBROUTINE - MSLOOP -
3210 ;*****
3211 ;* - TEST LOOP SUBROUTINE -
3212 ;* THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
3213 ;* TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD. THE
3214 ;* CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
3215 ;* DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
3216 ;* THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
3217 ;* ROUTINE AND THEN ONCE EACH MILLI-SECOND THEREAFTER.
3218 ;*
3219 ;* INPUTS: R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
3220 ;* R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
3221 ;* R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
3222 ;* R4 - ADDRESS OF THE WORD TO TEST.
3223 ;* MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
3224 ;*
3225 ;* OUTPUTS: CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
3226 ;*
3227 ;* CALLING SEQUENCE: JSR PC,MSLOOP
3228 ;*
3229 ;* COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
3230 ;* CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
3231 ;* ON THE SYSTEM.
3232 ;* THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
3233 ;* DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
3234 ;* LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
3235 ;* IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
3236 ;* THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
3237 ;* IF THE CONDITION IS MET, FAILURE OTHERWISE.
3238 ;*
3239 ;* SUBORDINATE ROUTINES CALLED: MSLGET.
3240 ;*****
3241
3242 015044 MSLOOP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
015044 004567 167010 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3243
3244 ;*
3245 ;* CALLING THE MSLGET ROUTINE FROM THE MSLOOP ROUTINE ISOLATES THE CALLER OF
3246 ;* MSLOOP FROM THE RETURNED TEST WORD AND REMAINING TIME-OUT VALUES.
3247 ;*
3248 015050 004767 177654 JSR PC,MSLGET ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.
3249
3250 015054 60$: PASS ;RESTORE GPRS,
015054 004736 JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
3251 015056 000207 RTS PC ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.

```

M7

CZDHVBO DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 70
GLOBAL SUBROUTINE - OOPS -

SEQ 90

```

3253 .SBTTL GLOBAL SUBROUTINE - OOPS -
3254 ;* *****
3255 ;* - PROGRAM ABORT SUBROUTINE -
3256 ;* THIS SUBROUTINE IS USED TO ABORT THE PROGRAM WHEN A FATAL ERROR IS
3257 ;* DETECTED IN THE PROGRAM OR THE HOST SYSTEM HARDWARE. AN ERROR MESSAGE
3258 ;* IS PRINTED GIVING SOME INFORMATION ABOUT THE NATURE OF THE ABORT.
3259 ;*
3260 ;* INPUTS: R1 - ERROR CODE GIVING REASON FOR ABORT.
3261 ;*
3262 ;* OUTPUTS: AN ERROR MESSAGE IS PRINTED.
3263 ;* A LIST OF RETURN PC VALUES FOR ALL SUBROUTINE CALLS IS PRINTED.
3264 ;*
3265 ;* CALLING SEQUENCE: JSR PC,OOPS
3266 ;*
3267 ;* COMMENTS:
3268 ;*
3269 ;* SUBORDINATE ROUTINES CALLED: NONE.
3270 ;* -- *****
3271
3272 015060 OOPS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3273 015060 004567 166774 ; JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3274 015064 ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
3275 015064 104454 ERRSF 101,EM0101
3276 015066 000145 TRAP C$ERSF
3277 015070 015124 .WORD 101
3278 015072 000000 .WORD EM0101
3279 .WORD 0
3280 ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
3281 PRINTF #EM0102
3282 015074
3283 015074 012746 015210 MOV #EM0102,-(SP)
3284 015100 012746 000001 MOV #1,-(SP)
3285 015104 010600 MOV SP,R0
3286 015106 104417 TRAP C$PRINTF
3287 015110 062706 000004 ADD #4,SP
3288 2$: BREAK ;LOOK FOR OPERATOR CONTROL-C INPUT.
3289 TRAP C$BRK
3290 BR 2$ ;INFINITE LOOP.
3291 60$: PASS ;DCN'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
3292 JSR PC,R(SP)+ ;RETURN TO PREG05 SUBRT.
3293 RTS PC ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.
3282 015124 110 117 123 EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
3283 015127 124 040 103
3284 015132 117 115 120
3285 015135 125 124 105
3286 015140 122 040 110
3287 015143 101 122 104
3288 015146 127 101 122
3289 015151 105 040 117
3290 015154 122 040 123
3291 015157 117 106 124
3292 015162 127 101 122
3293 015165 105 040 102
3294 015170 125 107 040
3295 015173 105 116 103
3296 015176 117 125 116
3297 015201 124 105 122

```

N7

CZDHVBO DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 70-1
GLOBAL SUBROUTINE - OOPS -

SEQ 91

	015204	105	104	056	
	015207	000			
3283	015210	045	116	045	EM0102:: ,ASCIZ /%N%APROGRAM HUNG, WAITING FOR A CONTROL-C. <*****N%N/
	015213	101	120	122	
	015216	117	107	122	
	015221	101	115	040	
	015224	110	125	116	
	015227	107	054	040	
	015232	127	101	111	
	015235	124	111	116	
	015240	107	040	106	
	015243	117	122	040	
	015246	101	040	103	
	015251	117	116	124	
	015254	122	117	114	
	015257	055	103	056	
	015262	040	074	052	
	015265	052	052	052	
	015270	052	052	052	
	015273	052	052	052	
	015276	052	052	052	
	015301	045	116	045	
3284	015304	116	000		

.EVEN

```

3286 .SBTTL GLOBAL SUBROUTINE - PRTLPR -
3287 ; *****
3288 ; -PRINT THE CONTENTS OF THE LPR.
3289 ; THIS ROUTINE IS USED TO PRINT OUT EXTENDED INFORMATION ON THE
3290 ; CONTENTS OF THE LINE PARAMETER REGISTER (LPR).
3291 ;
3292 ; INPUTS: R3 - CONTAINS THE NUMBER OF THE LINE YOU WISH TO EXAMINE.
3293 ; CSRA - CONTAINS THE ADDRESS OF THE DUT'S CSR.
3294 ; IESTAT - CONTAINS THE CURRENT STATUS OF THE TX AND RX INTERRUPT
3295 ; ENABLE BITS IN THE DUT'S CSR.
3296 ; LPRA - CONTAINS THE ADDRESS OF THE DUT'S LPR REGISTER.
3297 ;
3298 ; OUTPUTS: AN EXTENDED INFORMATION MESSAGE IS PRINTED ON THE OPERATORS
3299 ; CONSOLE.
3300 ;
3301 ; CALLING SEQUENCE: JSR PC,PRTLPR
3302 ;
3303 ; COMMENTS: THIS ROUTINE CHANGES THE INDIRECT ADDRESS FIELD OF THE DEVICE
3304 ; UNDER TEST'S CSR.
3305 ;
3306 ; SUBORDINATE ROUTINES CALLED: NONE.
3307 ; *****
3308
3309 015306 PRTLPR:SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3310 015306 004567 166546 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3311 015312 016701 164710 MOV CSRA,R1 ;GET THE CSR ADDRESS.
3312 015316 016702 164710 MOV LPRA,R2 ;GET THE LPR ADDRESS.
3313 015322 042703 177760 BIC #177760,R3 ;CLEAR ANY UNWANTED BITS.
3314 015326 056703 164724 BIS IESTAT,R3 ;SET STATE OF TX AND RX INTERRUPT ENABLE BITS.
3315 015332 010311 MOV R3,(R1) ;SELECT LINE.
3316 015334 011204 MOV (R2),R4 ;GET CONTENTS OF THE LPR.
3317 015336 ;PRINT MESSAGE"CONTENTS OF THE LPR:NNNNN"
3318 015336 010446 PRINTX #EF9019,#EM9026,R4;PRINT OUT MESSAGE ON OPERATORS CONSOLE.
3319 015340 012746 011100 MOV R4,-(SP)
3320 015344 012746 005034 MOV #EM9026,-(SP)
3321 015350 012746 000003 MOV #EF9019,-(SP)
3322 015354 010600 MOV #3,-(SP)
3323 015356 104415 MOV SP,R0
3324 015360 062706 000010 TRAP C#PNTX
3325 ADD #10,SP
3326
3327 604: PASS ;RESTORE GPRS.
3328 RTS PC JSR PC,@(SP) ;RETURN TO PREG05 SUBRT.

```

```

3321 .SBTTL GLOBAL SUBROUTINE - PUFIFO
3322 *****
3323 - PURGE THE FIFO
3324 THIS ROUTINE TRIES TO REMOVE ALL THE CHARACTERS FROM THE FIFO.
3325 ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE BMP CODE QUEUE.
3326
3327 INPUTS: RBUFA- CONTAINS THE ADDRESS OF THE RECEIVER.
3328
3329
3330 OUTPUTS: CARRY BIT - INDICATES THE STATE OF THE FIFO. SET: = PURGED.
3331 BMPCQ - THE CONTENTS OF THE BMP CODE QUEUE MAY BE UPDATED.
3332
3333 CALLING SEQUENCE: JSR PC,PUFIFO
3334
3335 COMMENTS:
3336
3337 SUBORDINATE ROUTINES CALLED: SAVBMP.
3338 *****
3339
3340 PUFIFO::SAVE JSR R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
3341 015370 004567 166464 ;CALL REGISTER SAVE SUBRT.
3342 015374 012701 001000 ;SET MAXIMUM TRY COUNT OF 512.
3343 015400 016704 164624 ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
3344
3345 2: MOV (R4),R2 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
3346 BPL 6: ;EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.
3347
3348 ; CHECK IF THE READ CHARACTER IS ACTUALLY A BMP CODE.
3349 ; IF IT IS, THEN SAVE IT ON THE BMP CODE QUEUE TO BE REPORTED LATER.
3350
3351 015410 012700 070000 ;GENERATE A BIT MAP OF CHAR ERROR BITS
3352 015414 040200 ; WHICH ARE NOT SET FOR CHAR.
3353 015416 001006 ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
3354
3355 ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
3356
3357 015420 012700 000301 ; CHECK IF BMP.
3358 015424 040200 ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
3359 015426 001002 ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
3360 015430 004767 001044 ;SAVE BMP CODE ON THE QUEUE.
3361
3362 4: DEC R1 ;DECREMENT THE TRY COUNT.
3363 BNE 2: ;LOOP TO TRY AGAIN.
3364 CLC ;CLEAR CARRY, TO INDICATE FIFO NOT PURGED.
3365 BR 60: ;EXIT WITH CARRY CLEAR.
3366
3367 6: SEC ;SET CARRY, TO INDICATE FIFO PURGED.
3368
3369 60: PASS ;RESTORE GPRS,
3370 015446 004736 JSR PC,0(SP); ;RETURN TO PREG05 SUBRT.
3371 RTS PC ;CARRY BIT, SET INDICATES FIFO PURGED.

```

GLOBAL SUBROUTINE

```

3371 .SBTTL GLOBAL SUBROUTINE - PUFIFR -
3372 *****
3373 ;* - PURGE FIFO REPORT ANY ERRORS FOUND.
3374 ;* THIS ROUTINE REMOVES ALL DATA FROM THE FIFO. ANY BMP CODES THAT ARE
3375 ;* FOUND ARE SAVE ON THE QUEUE TO BE REPORTED LATER IN THE BMP REPORT TEST.
3376 ;* ANY UNEXPECTED DATA (IE ANY NON-STATUS INFORMATION) THAT ARE FOUND,
3377 ;* ARE REPORTED AS AN ERROR.
3378 ;* IF THE FIFO WILL NOT PURGE AFTER 512 ATTEMPTS, THEN THE CURRENT TEST
3379 ;* THAT CALLED THIS ROUTINE RECEIVES A FAILURE FLAG THAT SHOULD BE USED
3380 ;* TO ABORT THE TEST.
3381 ;*
3382 ;* INPUTS: ERRIBL - ERRTYPE, ERMMSG, ERNBR ARE SET UP CORRECTLY.
3383 ;* RBUFA- CONTAINS THE ADDRESS OF THE RECEIVER.
3384 ;*
3385 ;* OUTPUTS: CARRY BIT - ABORT TEST FLAG, CLR = ABORT TEST, SET = OK.
3386 ;* ERBLK - VALUE WILL BE DESTROYED.
3387 ;* BMPCP - THE BMP CODE QUEUE POINTER MAY BE UPDATED.
3388 ;* THE CONTENTS OF THE BMP CODE QUEUE MAY BE UDATED.
3389 ;*
3390 ;* CALLING SEQUENCE: JSR PC,PUFIFR
3391 ;*
3392 ;* COMMENTS: THIS ROUTINE REPORTS ERRORS WITH NUMBERS INITIAL ERNBR
3393 ;* THRU TO ERNBR+2.
3394 ;* THE ERNBR IS RESTORED TO ITS INITIAL VALUE BEFORE RETURNING.
3395 ;*
3396 ;* SUBORDINATE ROUTINES CALLED: ER1603,ER9001,ER9002,SAVBMP.
3397 *****
3398
3399 015452 PUFIFR::SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3400 015452 004567 166402 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3401 015456 016746 166370 ;MOV ERRNBR,-(SP) ;SAVE THE CONTENTS OF THE ERROR NUMBER.
3402 015462 012705 001000 ;MOV #512,R5 ;SET MAXIMUM READ COUNTER TO 2*FIFO SIZE.
3403 ;*
3404 ;* READ DATA FROM THE FIFO UNTIL DATA VALID IS CLEAR OF READ COUNTER IS ZERO.
3405 ;* REPORT ANY BMP OR UNEXPECTED DATA AS ERRORS.
3406 015466 017702 164536 2: MOV RBUFA,R2 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
3407 015472 100063 ;LPL B ;EXIT IF DATA VALID CLEAR, IE. FIFO PURGED.
3408 ;*
3409 ;* CHECK IF READ DATA IS STATUS OR UNEXPECTED CHARACTER.
3410 ;*
3411 015474 012700 070000 ;MOV #70000,R0 ;GENERATE A BIT MAP OF CHAR ERROR BITS
3412 015500 040200 ;BIC R2,R0 ;WHICH ARE NOT SET FOR CHAR.
3413 015502 001012 ;BNE #4 ;SKIP BMP CHECK IF IT IS UNEXPECTED DATA.
3414 ;*
3415 ;* CHECK IF THE READ DATA IS MODEM STATUS, BMP OR SELFTEST?.
3416 ;* IF IT IS A BMP CODE THEN SAVE IT ON THE QUEUE.
3417 ;*
3418 015504 012767 012544 166344 ;MOV #ER9001,ERRBLK ;SET UP THE CORRECT ERROR REPORTING ROUTINE.
3419 015512 012700 000300 ;MOV #300,R0 ;CHECK IF BMP OR SELFTEST?.
3420 015516 040200 ;BIC R2,R0 ;TRY TO CLEAR BMP FLAGS IN THE READ DATA
3421 015520 001003 ;BNE #4 ;SKIP BMP ERROR REPORT IF MODEM OR SELFTEST?.
3422 015522 004767 000752 ;JSR PC,SAVBMP ;SAVE THE BMP CODE ON THE QUEUE.
3423 015526 000430 ;BR #6 ;BRANCH TO CHECK READ COUNT.
3424 ;*
3425 ;* CHECK IF THE READ DATA IS MODEM, SELFTEST OR UNEXPECTED DATA.
3426 ;*

```

E8

CZDHY80 DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 73-1
GLOBAL SUBROUTINE - PUFIFR -

SEN 95

```

3427 015530 032702 000001      4$: BIT    #BIT0,R2      ;TEST THE MODEM STATUS INDICATION BIT.
3428 015534 001425      BEQ    6$      ;DO NOT REPORT ANY ERROR IF MODEM STATUS.
3429 015536 012701 011124      MOV    #EM9104,R1      ;PASS THE CORRECT ERROR MESSAGE TO REPORT.
3430 015542 010203      MOV    R2,R3      ;EXTRACT THE LINE NUMBER FROM
3431 015544 000303      SWAB   R3      ; THE READ DATA.
3432 015546 042703 177760      BIC    #177760,R3      ;
3433 015552 006303      ASL    R3      ;FORM LINE NUMBER TIMES 2 FOR ER9002 ROUTINE.
3434 015554 052704 100000      BIS    #BIT15,R4      ;SET THE "NONE" EXPECTED MESSAGE FLAG.
3435 015560 005267 166266      INC    ERRNBR      ;SET ERROR NUMBER TO INITIAL ERRNBR+1.
3436 015564 012767 012644 166264      MOV    #ER9002,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
3437      ;REPORT ERROR "UNEXPECTED DATA FOUND IN FIFO".
3438 015572      ERROR      ;
3439      015572 104460      ;>>>> ERROR <<<<<.
3440      ;TRAP C$ERROR
3441      ;
3442      ;EXIT WITH FAILURE IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
3443      ;
3444      BIT    #BIT06,OPTION      ;EXIT WITH TEST FAILURE MESSAGE IF
3445      BEQ    7$      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
3446      ;DURING THE SOFTWARE QUESTIONS.
3447      DEC    ERRNBR      ;RESTORE ERROR NUMBER TO INITIAL ERRNBR.
3448      6$: DEC    R5      ;DECREMENT READ COUNTER.
3449 015612 001325      BNE    2$      ;LOOP TO READ NEXT CHAR FROM FIFO IF COUNT > 0.
3450      ;
3451      ;THE FIFO WILL NOT CLEAR, REPORT THE ERROR AND INDICATE THAT THE TEST IS TO
3452      ;BE ABORTED.
3453      ;
3454 015614 062767 000002 166230      ADD    #2,ERRNBR      ;SET ERROR NUMBER TO INITIAL ERRNBR+2.
3455 015622 012767 012020 166226      MOV    #ER1603,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
3456 015630 012701 010767      MOV    #EM9017,R1      ;PASS THE MESSAGE TO BE REPORTED.
3457      ;REPORT THE ERROR "FIFO WILL NOT PURGE, (DATA VALID STUCK SET)"
3458      ;"?????? TEST ABORTED".
3459 015634      ERROR      ;>>>> ERROR <<<<<.
3460      015634 104460      ;TRAP C$ERROR
3461 015636 000241      7$: CLC      ;INDICATE THE TEST IS TO BE ABORTED.
3462 015640 000401      BR     10$      ;EXIT THIS ROUTINE AND ABORT THE CURRENT TEST.
3463 015642 000261      8$: SEC      ;SET THE CARRY, DO NOT ABORT THE TEST.
3464      ;
3465 015644 012667 166202      10$: MOV    (SP)+,ERRNBR      ;RESTORE INITIAL ERROR NUMBER.
3466 015650      60$: PASS      ;RESTORE GPRS.
3467      ;JSR      PC,#(SP)+      ;RETURN TO PREG05 SUBRT.
3468      ;CARRY BIT, SET INDICATES FIFO PURGED, DO NOT
3469 015652 000207      RTS    PC      ;ABORT THE TEST.

```

```

3471      ,SBTTL GLOBAL SUBROUTINE - READBX -
3472      ;* *****
3473      ;* - READ CHARACTERS FROM THE FIFO AND CHECKS FOR BMPS AND XONS-
3474      ;* THIS SUBROUTINE IS USED IN THE FIMAVL.TST.
3475      ;* IT READS THE SPECIFIED NUMBER OF CHARACTERS FROM THE FIFO AND CHECKS
3476      ;* FOR BMP CODES AND XON CHARACTERS.
3477      ;*
3478      ;* INPUTS:      R0 - CONTAINS THE NUMBER OF CHARS TO READ FROM THE FIFO.
3479      ;*
3480      ;* OUTPUTS:     R1 - CONTAINS ADDRESS OF ERROR MESSAGE TO BE REPORTED
3481      ;*              CLEAR IF NO ERROR FOUND.
3482      ;*              CARRY USED TO INDICATE IF FIFO WAS FOUND EMPTY, CARRY CLEAR.
3483      ;*
3484      ;* CALLING SEQUENCE: JSR PC,READ
3485      ;*
3486      ;* COMMENTS:
3487      ;*
3488      ;* SUBORDINATE ROUTINES CALLED: CHKBMP.
3489      ;* - *****
3490
3491      READBX:: SAVE          JSR      ;SAVE CONTENTS OF GPRS R0 THRU R5.
3492      015654 004567 166200      R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3493      015660 005001          CLR     R1    ;CLEAR GPR THAT HOLDS THE ADDRESS OF ERRMSG.
3494      015662 016703 164342      MOV     RBUFA,R3 ;GET THE ADDRESS OF THE RECEIVER BUFFER REG.
3495      015666 011302      2$:      MOV     (R3),R2 ;READ A CHARACTER FROM THE FIFO.
3496      015670 100015      BPL      B$      ;BRANCH IF FIFO IS EMPTY.
3497      ;*
3498      ;* CHECK IF THE READ CHARACTER IS A BMP CODE.
3499      ;* IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
3500      ;* ABORT THE TEST.
3501      ;*
3502      JSR     PC,CHKBMP          ;CHECK IF CHARACTER IS A BMP CODE.
3503      BCS     6$                ;BRANCH IF A BMP CODE WAS FOUND.
3504      CMPB    R2,021            ;CHECK IF IT IS AN XON.
3505      BNE     4$                ;BRANCH IF NOT AN XON.
3506      MOV     @EM5402,R1        ;PASS THE MESSAGE TO BE REPORTED.
3507      BR      6$                ;GO EXIT TEST.
3508      015714 005300      4$:      DEC     R0    ;DECREMENT THE READ COUNT.
3509      015716 001363      BNE     2$
3510      015720 000261      6$:      SEC
3511      015722 000401      BR      60$      ;SET CARRY TO INDICATE SUCCESS.
3512      015724 000241      8$:      CLC          ;EXIT
3513      015726          60$:      PASS    R1      ;CLEAR CARRY BIT TO INDICATE FAILURE.
3514      015726 010166 000004      MOV     R1,R1SLOT(SP) ;RESTORE GPRS,
3515      015732 004736          JSR      PC,@(SP)+ ;PUT R1 IN STACK SLOT.
3516      015734 000207          RTS     PC      ;RETURN TO PREG05 SUBRT.

```

```

3516 .SBTTL GLOBAL SUBROUTINE - REPDER -
3517 ;+ *****
3518 ;* - REPORT DATA ERRORS -
3519 ;* THIS SUBROUTINE IS USED TO REPORT INCORRECT CHARACTERS AND LINE
3520 ;* NUMBERS IN A WORD OF DATA READ FROM THE RXFIFO. THIS ROUTINE
3521 ;* CHECKS FOR THE NUMBER OF INDIVIDUAL DATA ERRORS ON A LINE EXCEEDING
3522 ;* THE REQUESTED AMOUNT AND SETS THE APPROPRIATE ERROR SUMMARY FLAG. IT
3523 ;* THEN STOPS REPORTING ANY FURTHER ERRORS ON THAT LINE. ANY BMP CODES
3524 ;* FOUND ARE SAVED ON THE BMP CODE QUEUE TO BE REPORTED LATER AND ANOTHER
3525 ;* CHARACTER IS READ FROM THE RXFIFO.
3526 ;*
3527 ;* INPUTS: R1 - CONTAINS THE NUMBER OF THE UUT.
3528 ;* R3 - LOW BYTE CONTAINS THE EXPECTED CHAR.
3529 ;* ERCNTB - ADDRESS OF THE BASE OF THE ERROR SUMMARY TABLE.
3530 ;* ERRBLK - ADDRESS OF ERROR REPORTING ROUTINE IN ERROR TABLE.
3531 ;* ERRNBR - SET TO THE ERROR NUMBER OF THE FIRST ERROR IN THIS ROUTINE.
3532 ;* EXOERR - ADDRESS OF "EXIT ON ERROR" FLAG.
3533 ;*
3534 ;* OUTPUTS: CARRY - CLEAR IF RXFIFO WAS EMPTY, SET OTHERWISE.
3535 ;* ERCNTB - CONTENTS UPDATED TO REFLECT THE CURRENT STATE.
3536 ;* ERRBLK - CONTENTS MAY BE ALTERED.
3537 ;* ERSMPF - ERROR SUMMARY FLAGS UPDATED.
3538 ;* EXOERR - 1 IF AT LEAST ONE ERROR OCCURED.
3539 ;* 0 IF NO ERRORS OCCURED.
3540 ;*
3541 ;* CALLING SEQUENCE: JSR PC,REPDER
3542 ;*
3543 ;* COMMENTS: THIS ROUTINE REPORTS ERRORS WITH NUMBERS INITIAL ERRNBR
3544 ;* THRU INITIAL ERRNBR+1.
3545 ;*
3546 ;* SUBORDINATE ROUTINES CALLED: NONE
3547 ;+ *****
3548
3549 015736 REPDER:: SAVE
3550 015736 004567 166116 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3551 015742 005067 000304 CLR 62$ ;CLEAR THE "AN ERROR OCCURED" INDICATOR.
3552 015746 012705 000200 MOV 0128.,R5 ;SET THE MAX BMP READ COUNT.
3553 015752 000241 2$: CLC ;CLEAR THE CARRY IN CASE THE FIFO IS EMPTY.
3554 015754 017702 164250 MOV 08BUFA,R2 ;READ THE RXFIFO.
3555 015760 100132 BPL 61$ ;EXIT THIS SUBROUTINE WITH CARRY CLEAR IF FIFO EMPTY.
3556 ;*
3557 ;* CHECK IF THE CHARACTER IS A BMP CODE.
3558 015762 012700 170301 MOV 0170301,R0 ;SET UP THE BIT MASK FOR A BMP CODE.
3559 015766 040200 BIC R2,R0 ;TRY AND CLEAR THE BIT MASK.
3560 015770 001004 BNE 4$ ;AVOID SAVING THE CODE IF IT IS NOT A BMP CODE.
3561 015772 004767 000502 JSR PC,SAVBMP ;SAVE THE BMP CODE ON THE QUEUE.
3562 015776 005305 DEC R5 ;DECREMENT THE MAX BMP READ COUNT.
3563 016000 001364 BNE 2$ ;READ ANOTHER CHARACTER FROM THE RXFIFO.
3564 ;*
3565 ;* VERIFY THAT THE READ CHARACTER IS CORRECT.
3566 4$: MOV R2,R5 ;SAVE THE READ DATA.
3567 016002 010205 BIC 0177400,R2 ;CLEAR THE CLUTTER FROM THE DATA CHAR.
3568 016004 042702 177400 CMPB R2,R3 ;COMPARE THE READ CHAR WITH THE EXPECTED CHAR.
3569 016010 120203 BEQ 6$ ;AVOID THE ERROR REPORT IF THE DATA IS CORRECT.
3570 016012 001432 MOV 01,62$ ;INDICATE AN ERROR HAS OCCURED.
3571 016014 012767 000001 000230

```


3627	016234	101003		BHI	60\$		;AVOID SETTING THE ERROR SUMMARY FLAG IF THE
3628							;NUMBER OF ERRORS REPORTED IS LESS THAN THE
3629							;NUMBER REQUESTED.
3630	016236	056467	002336	164346	BIS	BITTBL(R4),ERSMRF	;OTHERWISE SET THE APPROIATE ERROR FLAG.
3631							
3632	016244	000261		60\$:	SEC		;SET THE CARRY FLAG AND INDICATE SUCCESS.
3633	016246			61\$:	PASS		
	016246	004736				JSR	PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
3634	016250	000207			RTS	PC	
3635	016252	000000		62\$:	.WORD	0	;STORAGE FOR THE "AT LEAST ONE ERROR" INDICATOR.

```

3637 .SBTTL GLOBAL SUBROUTINE - REPSMR -
3638 ;* *****
3639 ;* - REPORT ERROR SUMMARY ROUTINE -
3640 ;* THIS SUBROUTINE REPORTS AN ERROR SUMMARY FOR THOSE LINES WHICH HAVE
3641 ;* EXCEEDED THE NUMBER OF INDIVIDUAL ERRORS TO REPORT FOR A SINGLE LINE
3642 ;* IN A SINGLE TEST. THIS PARAMETER CAN BE SPECIFIED BY THE OPERATOR IF
3643 ;* HE/SHE ANSWERS THE SOFTWARE PARAMETER QUESTIONS.
3644 ;*
3645 ;* INPUTS: ERCNTB - LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
3646 ;* ERRMSG - ADDRESS OF PRIMARY ERROR MESSAGE.
3647 ;* ERNBR - ERROR NUMBER OF ERRORS IN THIS ROUTINE.
3648 ;* ERSRPF - "REPORT ERROR SUMMARY FOR LINE" FLAGS.
3649 ;*
3650 ;* OUTPUTS: ERRBLK - ADDRESS OF ERROR REPORTING ROUTINE (DESTROYED).
3651 ;* SUMMARY MESSAGES MAY BE PRINTED AT THE OPERATOR CONSOLE.
3652 ;*
3653 ;* CALLING SEQUENCE: JSR PC,REPSMR
3654 ;*
3655 ;* COMMENTS: IF NO LINES HAVE EXCEEDED THE MAXIMUM NUMBER OF INDIVIDUAL
3656 ;* ERRORS TO REPORT, NO MESSAGES ARE PRINTED BY THIS ROUTINE.
3657 ;* ERROR SUMMARIES IN THIS ROUTINE ARE REPORTED AS ERRORS.
3658 ;* THE CONTENTS OF ERRBLK ARE DESTROYED.
3659 ;*
3660 ;* SUBORDINATE ROUTINES CALLED:
3661 ;* - *****
3662
3663 016254 REPSMR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3664 016254 004567 165600 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3665 016260 005767 164326 TST ERSRPF ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
3666 016264 001404 BEQ 60$ ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
3667 ;*
3668 ;* WE HAVE SOME ERROR SUMMARIES TO REPORT.
3669 016266 012767 013022 165562 MOV #ER9004,ERRBLK ;SELECT ERROR REPORTING ROUTINE.
3670 ;*
3671 ;* REPORT
3672 ;* "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
3673 ;*
3674 016274 ERROR
3675 016274 104460 TRAP C$ERROR
3676 016276 60$: PASS ;RESTORE GPRS.
3677 016276 004736 JSR PC,R(SP)+ ;RETURN TO PREG05 SUBRT.
3677 016300 000207 RTS PC

```

```

3679      ,SBTTL  GLOBAL SUBROUTINE          - RESETT -
3680      ;*****
3681      ;*          - RESET DEVICE UNDER TEST -
3682      ;*          THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
3683      ;*          IF RESET DOES NOT SUCCESSFULLY COMPLETE, IE. TIME-OUT OCCURS, THEN
3684      ;*          AN ABORT TEST ERROR MESSAGE IS REPORTED.
3685      ;*
3686      ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR
3687      ;*              TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
3688      ;*              ERRCTL - ERRTP,ERNBR,AND ERRMSG SET UP CORRECTLY.
3689      ;*
3690      ;* OUTPUTS:     THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
3691      ;*              CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
3692      ;*              ERRBLK - VALUE MAY BE DESTROYED.
3693      ;*              IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
3694      ;*              TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
3695      ;*
3696      ;* CALLING SEQUENCE:  JSR      PC,RESETT
3697      ;*
3698      ;* COMMENTS:      THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERNBR
3699      ;*                  THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERNBR.
3700      ;*
3701      ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET.
3702      ;*****
3703
3704      016302      RESETT:: SAVE                      ;SAVE CONTENTS OF GPRS R0 THRU R5.
3705      016302      004567 165552                      JSR      R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3706      016306      012702 000040                      MOV      #BIT05,R2 ;SET BIT MASK OF MASTER RESET BIT.
3707
3708      ;*
3709      ;* TEST THE STATE OF THE MASTER RESET BIT IN THE CSR.
3710      ;* IF MR IS SET THEN WAIT FOR SELF-TEST TO COMPLETE.
3711      ;* IF TIME-OUT OCCURS, REPORT THE ERROR AND PASS-OUT ABORT TEST INDICATOR.
3712      ;*
3713      016312      016704 163710                      MOV      CSRA,R4 ;GET THE ADDRESS OF THE DUT'S CSR.
3714      016316      030214                      BIT      R2,(R4) ;CHECK STATE OF MASTER RESET BIT.
3715      016320      001406                      BEQ      2$ ;DON'T DELAY IF MR IS ALREADY CLEAR.
3716      016322      005003                      CLR      R3 ;SET UP DESIRED STATE OF MASTER RESET BIT.
3717      016324      012701 011610                      MOV      #5000.,R1 ;PASS TIME-OUT VALUE OF 5 SECONDS.
3718      016330      004767 176374                      JSR      PC,MSLGET ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
3719      016334      103012                      BCC      4$ ;GO REPORT ERROR IF TIMEOUT OCCURRED.
3720
3721      ;*
3722      ;* SET MASTER RESET BIT IN CSR. CLEAR TX AND RX ENABLE BITS, ETC.
3723      ;* SKIP THE SELFTEST.
3724      ;* TIME-OUT OF 5 SECS, JUST IN CASE THE SELF-TEST EXECUTES.
3725      ;*
3726      016336      010277 163664                      2$: MOV      R2,0CSRA ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
3727      016342      004767 000246                      JSR      PC,SKPSTS ;TRY TO SKIP THE SELFTEST.
3728
3729      ;*
3730      ;* SET SELF-TEST TIME-OUT OF 5 SECONDS, AND WAIT FOR M.R TO CLEAR.
3731      ;* IF TIME-OUT OCCURS, THEN REPORT THE FATAL ERROR AND PASS-OUT THE ABORT
3732      ;* TEST INDICATOR.
3733      ;*
3734      016346      005003                      CLR      R3 ;SET UP DESIRED STATE OF MASTER RESET BIT.
3735      016350      012701 011610                      MOV      #5000.,R1 ;PASS TIME-OUT VALUE OF 5 SECONDS.
3736      016354      004767 176350                      JSR      PC,MSLGET ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
3737      016360      103410                      BCS      6$ ;SKIP ERROR REPORT IF MR CLEARED IN TIME.

```

L8

CZDHV80 DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 77-1
GLOBAL SUBROUTINE - RESETT -

SEQ 102

```

3735
3736      ; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET,TEST ABORTED".
3737      ; INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
3738      ;-
3739 016362 012701 005320      4$:  MOV     $EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
3740 016366 012767 012020 165462  MOV     $ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
3741      ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
3742      ; "TEST ABORTED"
3743 016374      ERROR                                ; >>>>> ERROR <<<<<
3744 016374 104460      TRAP      C$ERROR
3745 016376 000241      CLC                                ;INDICATE TEST IS TO BE ABORTED.
3746 016400 000403      BR      60$                        ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
3747      ;-
3748      ; CLEAR TX AND RX INTERRUPT ENABLE STATUS FLAGS IN IESTAT.
3749      ; EXIT WITH CONTINUE TEST INDICATOR SET (IE,CARRY SET).
3750 016402 005067 163650 6$:  CLR      IESTAT          ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
3751 016406 000261      SEC                                ;INDICATE SUCCESS, CONTINUE TEST.
3752
3753 016410      60$:  PASS                                ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
3754 016410 004736      JSR      PC,B(SP)+                ;RETURN TO PREGOS SUBRT.
3755 016412 000207      RTS      PC                      ;CARRY BIT;IF CLEAR,INDICATES ABORT TEST.
3756

```

```

3758 .SBTTL GLOBAL SUBROUTINE - RXIEO -
3759 ;* *****
3760 ;* - RECEIVER INTERRUPT DISABLE -
3761 ;* THIS ROUTINE IS USED TO DISABLE RECEIVER INTERRUPTS IN THE DHU11.
3762 ;*
3763 ;* INPUTS: NONE.
3764 ;*
3765 ;* OUTPUTS: THE RX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
3766 ;* IESTSI -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3767 ;* ENABLE BITS.
3768 ;*
3769 ;* CALLING SEQUENCE: JSR PC,RXIEO
3770 ;*
3771 ;* COMMENTS: THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3772 ;* THE DUT CSR ARE DESTROYED.
3773 ;*
3774 ;* SUBORDINATE ROUTINES CALLED: NONE.
3775 ;* *****
3776 RXIEO:: MOV RO,-(SP) ;SAVE CONTENTS OF RO ON THE STACK.
3777 GETPRI -(SP) ;SAVE PROCESSOR PRIORITY ON STACK.
3778 016414 010046
3779 016416 104440
3780 016420 010046
3781 016422 SETPRI @PRI07 ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
3782 016422 012700 000340
3783 016426 104441
3784 016430 042767 137777 163620
3785 016436 016777 163614 163562
3786 016444
3787 016444 012600
3788 016446 104441
3789 016450 012600
3790 016452 000207
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000

```

N8

CZDHVBO DHU-11 FUNC TST PART2
GLOBAL SUBROUTINE

MACRO M1200 15-MAR-84 09:28 PAGE 79
- RXIE1 -

SEQ 104

```

3785      .SBTTL GLOBAL SUBROUTINE                      - RXIE1 -
3786      ; * *****
3787      ; * - RECEIVER INTERRUPT ENABLE -
3788      ; * THIS ROUTINE IS USED TO ENABLE RECEIVER INTERRUPTS IN THE DHU11.
3789      ; *
3790      ; * INPUTS:      NONE.
3791      ; *
3792      ; * OUTPUTS:     THE RX.INT.ENBL BIT IS SET IN THE DUT CSR.
3793      ; *               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3794      ; *               ENABLE BITS.
3795      ; *
3796      ; * CALLING SEQUENCE:  JSR      PC,RXIE1
3797      ; *
3798      ; * COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3799      ; *               THE DUT CSR ARE DESTROYED.
3800      ; *
3801      ; * SUBORDINATE ROUTINES CALLED: NONE.
3802      ; * - *****
3803
3804 016454 052767 000100 163574 RXIE1:: BIS      #BIT06,IESTAT ;SET RX.INT.ENBL BIT IN IESTAT.
3805 016462 042767 137677 163566      BIC      #137677,IESTAT ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
3806 016470 016777 163562 163530      MOV      IESTAT,#CSRA  ;ENABLE RX INTERRUPTS.
3807 016476 000207          RTS      PC

```

```

3809 .SBTTL GLOBAL SUBROUTINE - SAVBMP -
3810 ;* *****
3811 ;* - SAVE BMP CODES ROUTINE -
3812 ;* THIS ROUTINE SAVES THE PARAMETER PASSED IN, ONTO THE BMP CODE QUEUE
3813 ;* TOGETHER WITH THE NUMBER OF THE CURRENTLY EXECUTING TEST.
3814 ;*
3815 ;* INPUTS: R2 - CONTAINS THE BMP CODE THAT IS TO BE PLACED ON THE QUEUE.
3816 ;* BMPCQP - CONTAINS ADDRESS OF NEXT LOCATION IN THE BMP QUEUE.
3817 ;* BMPCQB - LABEL AT BASE OF THE BMP CODE QUEUE.
3818 ;* BMPCQE - LABEL OF NEXT LOCATION AFTER THE END OF THE BMP QUEUE.
3819 ;* TSTNUM - CONTAINS THE NUMBER OF THE CURRENT TEST.
3820 ;*
3821 ;* OUTPUTS: BMPCQP - INCREMENTED BY 4.
3822 ;* THE CONTENTS OF THE BMP CODE QUEUE ARE UPDATED.
3823 ;*
3824 ;* CALLING SEQUENCE: JSR PC,SAVBMP
3825 ;*
3826 ;* COMMENTS: IF THE OVERFLOW OCCURS THEN THE LAST LOCATION WILL BE
3827 ;* OVERWRITTEN BY ANY SUBSEQUENT ATTEMPTS TO UPDATE THE QUEUE.
3828 ;*
3829 ;* SUBORDINATE ROUTINES CALLED: NONE.
3830 ;* - *****
3831
3832 016500 SAVBMP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3833 016500 004567 165354 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3834 016504 016704 163700 ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
3835 016510 116724 163540 ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
3836 016514 003204 ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
3837 016516 042702 177400 ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
3838 016522 010224 ;SAVE THE BMP CODE ON THE QUEUE.
3839 016524 020427 002612 ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
3840 016530 103402 ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
3841 016532 162704 000004 ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
3842 016536 010467 163646 ;SAVE THE POINTER.
3843 016542 ;
3844 016542 004736 ;RESTORE GPRS,
016544 000207 ;PC,0(SP). ;RETURN TO PREG05 SUBRT.
RTS PC JSR

```

```

3846      .SBTTL GLOBAL SUBROUTINE                      - SETPAR
3847      ;* *****
3848      ;* - SET TX AND CONTROL PARAMETERS -
3849      ;* THIS SUBROUTINE IS USED IN THE FIHAVL.TST.
3850      ;* IT INITIALISES THE SELECTED LINE TO THE FOLLOWING STATE:
3851      ;* INTERNAL LOOPBACK, IAUTO ENABLED, LPR:38.4K, 8 BITS/CHAR, 2 STOP,
3852      ;* ODD PARITY.
3853      ;*
3854      ;* INPUTS:      R1 - CONTAINS NUMBER OF THE LINE TO BE INITIALISED.
3855      ;*
3856      ;* OUTPUTS:     LNCTRL AND LPR REGISTERS FOR THE SELECTED LINE ARE DESTROYED.
3857      ;*
3858      ;* CALLING SEQUENCE: JSR      PC,SETPAR
3859      ;*
3860      ;* COMMENTS:
3861      ;*
3862      ;* SUBORDINATE ROUTINES CALLED: DELAY,WTWLNC,WTWLPR.
3863      ;* - *****
3864
3865      SETPAR:: SAVE
3866      016546 004567 165306      JSR      PC,LINBIT      ;SAVE CONTENTS OF GPRS R0 THRU R5.
3867      016552 004767 176072      MOV      R0,R5          R5,PREG05      ;CALL REGISTER SAVE SUBRT.
3868      016560 012700 000206      MOV      #206,R0        ;GET A BIT MAP FOR THIS LINE.
3869      016564 004767 000762      JSR      PC,WTWLNC        ;COPY THE LINE BIT MAP.
3870      016570 012700 177670      MOV      #177670,R0     ;PASS INTERNAL LOOPBACK, ENABLE RX AND IAUTO.
3871      016574 004767 001002      JSR      PC,WTWLPR        ;INITIALISE THE LINE CONTROL REGISTER.
3872      016600 012704 000012      MOV      #10.,R4        ;PASS THE LPR CONTENTS.
3873      016604 004767 175452      JSR      PC,DELAY        ;SET THE LPR CONTENTS TO 38.4K BAUD.
3874      ;*                                     ;PASS DELAY TIME OF 10 MILLI SECONDS.
3875      016610 004736      601: PASS      JSR      PC,DELAY  ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
3876      016612 000207      RTS      PC                    ;RESTORE GPRS,
3877                                     ;PC,R(SP).          ;RETURN TO PREG05 SUBRT.

```

GLOBAL SUBROUTINE

SKPSTS

```

3878      SBTTL GLOBAL SUBROUTINE - SKPSTS -
3879      ;* *****
3880      ;* - SKIP SELFTEST ROUTINE -
3881      ;* THIS SUBROUTINE IS USED TO SKIP THE SELFTEST AFTER A DUT RESET HAS BEEN
3882      ;* INITIATED. IT MUST BE ENTERED IMMEDIATELY AFTER SETTING THE DUT MASTER
3883      ;* RESET ROUTINE OR AFTER THE EXECUTION OF A BUS RESET (BECAUSE OF TIMING
3884      ;* CONSIDERATIONS).
3885      ;*
3886      ;* INPUTS:      CSRA - CONTAINS ADDRESS OF THE DUT CSR.
3887      ;*              TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
3888      ;*
3889      ;* OUTPUTS:     SKIP SELFTEST CODES ARE WRITTEN TO THE DUT REGISTERS.
3890      ;*
3891      ;* CALLING SEQUENCE: JSR PC,SKPSTS
3892      ;*
3893      ;* COMMENTS:
3894      ;*
3895      ;* SUBORDINATE ROUTINES CALLED: DELAY.
3896      ;* - *****
3897
3898      SKPSTS:: SAVE          ;SAVE CONTENTS OF GPRS R0 THRU R5.
3899      ;*              JSR      R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3900      ;*              MOV      #10,R4    ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
3901      ;*              JSR      PC,DELAY   ;DELAY FOR 10 MILLI-SECONDS.
3902      ;*
3903      ;* WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
3904      ;* -
3905      ;* MOV      #NUMLNS!BIT05,R1 ;FORM IND.ADR.REG FIELD (PLUS M.R. BIT) WORD.
3906      ;* ;THE ABOVE INCLUSION OF THE M.R. BIT IS NECESSARY BECAUSE OF THE
3907      ;* ;LACK OF A M.R. BIT WRITE LOCK-OUT ON THE DHU-11.
3908      ;* MOV      #52525,R3        ;INITIALISE THE SKIP SELF-TEST CODE.
3909      ;* 4$:      DEC      R1        ;SELECT THE NEXT SET OF DEVICE REGISTERS.
3910      ;*          MOV      CSRA,R4    ;GET THE ADDRESS OF THE CSR OF THE DUT.
3911      ;*          MOV      R1,(R4)+   ;SELECT A BANK OF DUT REGISTERS.
3912      ;*          MOV      R3,(R4)+   ;WRITE THE CODE TO A DUT REGISTER.
3913      ;*          CMP      R4,TXBFCA  ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
3914      ;*          BLO      6$         ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
3915      ;*          BIT      #17,R1     ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
3916      ;*          BNE      4$         ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.
3917      ;* 60$:     PASS
3918      ;*          RTS      PC        ;RESTORE GPRS.
3919      ;*          JSR      PC,@(SP)+ ;RETURN TO PREG05 SUBRT.

```

```

3920 .SBTTL GLOBAL SUBROUTINE - TSABRT -
3921 ;* *****
3922 ;* - TEST ABORT ROUTINE -
3923 ;* THIS SUBROUTINE IS USED WHEN A NON-TEST RELATED ERROR HAS BEEN FOUND
3924 ;* DURING THE EXECUTION OF THE CURRENT TEST.
3925 ;* IT IS USED TO INFORM THE OPERATOR THAT THE CURRENT TEST HAS BEEN
3926 ;* ABORTED.
3927 ;*
3928 ;* INPUTS: ERRMSG - CONTAINS THE NAME OF THE CURRENT TEST.
3929 ;*          ERRNR - CONTAINS THE CORRECT ERROR NUMBER.
3930 ;*          THE REMAINDER OF THE ERRBL IS CORRECTLY INITIALISED.
3931 ;*
3932 ;* OUTPUTS: MESSAGES ARE REPORTED TO THE OPERATOR.
3933 ;*
3934 ;* CALLING SEQUENCE: JSR PC,TSABRT
3935 ;*
3936 ;* COMMENTS:
3937 ;*
3938 ;* SUBORDINATE ROUTINES CALLED: ER1603.
3939 ;* *****
3940
3941 016672 TSABRT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3942 016672 004567 165162 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3943 016676 012701 016714 MOV #24,R1 ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
3944 016702 012767 012020 165146 MOV #ER1603,ERRBLK ;SET-UP THE ERROR REPORTING ROUTINE.
3945 016710 104460 ;
3946 016712 000432 ; >>>>> ERROR <<<<<. TRAP C$ERROR
3947 016714 040 116 117 24: BR 60$ ;
3948 016717 116 055 122 .ASCIZ / NON-RELATED TEST ERROR FOUND DURING TEST EXECUTION/
3949 016722 105 114 101
3950 016725 124 105 104
3951 016730 040 124 105
3952 016733 123 124 040
3953 016736 105 122 122
3954 016741 111 122 040
3955 016744 106 117 125
3956 016747 116 104 040
3957 016752 104 125 122
3958 016755 111 116 107
3959 016760 040 124 105
3960 016763 123 124 040
3961 016766 105 130 105
3962 016771 103 125 124
3963 016774 111 117 116
3964 016777 000
3965
3966 .EVEN
3967 60$: PASS
3968 017000 004736 JSR ;RESTORE GPRS.
3969 017002 000207 RTS PC ;RETURN TO PREG05 SUBRT.

```

```

3951      .SBI L GLOBAL SUBROUTINE - TXDATP -
3952      ;++ *****
3953      ;* - TRANSMIT DATA PATTERN -
3954      ;* THIS SUBROUTINE IS USED IN THE FIHAVL.TST.
3955      ;* IT TRANSMITS A SPECIFIED NUMBER OF DATA BYTES ON THE SPECIFIED LINE.
3956      ;*
3957      ;* INPUTS:      R0 - CONTAINS THE NUMBER OF DATA BYTES TO TX.
3958      ;*              R1 - CONTAINS LINE NUMB ON WHICH TRANSMISSION IS TO TAKE PLACE.
3959      ;*              BUFBAS TO BUFMID CONTAINS A 256 BYTE DATA PATTERN.
3960      ;*
3961      ;* OUTPUTS:     DATA IS SENT OUT ON THE SPECIFIED LINE.
3962      ;*              CARRY SET = TX SUCCESSFUL.
3963      ;*
3964      ;* CALLING SEQUENCE: TXDATP
3965      ;*
3966      ;* COMMENTS:
3967      ;*
3968      ;* SUBORDINATE ROUTINES CALLED: DODMA.
3969      ;-- *****
3970
3971      TXDATP:: SAVE
3972      017004 004567 165050      MOV R0,R3      JSR      ;SAVE CONTENTS OF GPRS R0 THRU R5.
3973      017010 010003      MOV 48BUFBAS,R2      R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3974      017016 004767 175356      JSR PC,DODMA      ;PASS THE NUMBER OF CHARS TO TX.
3975      017022      PASS      ;PASS THE START OF THE DATA PATTERN TO TX.
3976      017024 000207      RTS PC      ;TRANSMIT THE DATA PATTERN.
3977      ;RESTORE GPRS.
3978      ;RETURN TO PREG05 SUBRT.

```

```

3978 .SBTTL GLOBAL SUBROUTINE - TXDSBL -
3979 ;* *****
3980 ;* - TRANSMITTER DISABLE -
3981 ;* THIS SUBROUTINE IS USED TO DISABLE TRANSMISSION ON SELECTED LINES BY,
3982 ;* CLEARING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
3983 ;*
3984 ;* INPUTS: R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO CLEAR TX.ENABLE.
3985 ;* CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
3986 ;* IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
3987 ;* NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
3988 ;* TXAD2A - CONTAINS THE ADDRESS OF THE TBUFFAD2 REGISTER.
3989 ;*
3990 ;* OUTPUTS: R5 - BIT'S SET INDICATE THE INITIAL STATES OF ALL TX.ENABLE BITS.
3991 ;* TBUFFAD2 - THE STATE OF THE TX.ENABLE BIT MAY BE ALTERED.
3992 ;* THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
3993 ;*
3994 ;* CALLING SEQUENCE: JSR PC,TXDSBL
3995 ;*
3996 ;* COMMENTS:
3997 ;*
3998 ;* SUBORDINATE ROUTINES CALLED: NONE.
3999 ;* - *****
4000
4001 TXDSBL:: SAVE R5,R0 THRU R5.
4002 017026 004567 165026 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4003 017032 010500 MOV R5,R0 ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
4004 017034 012701 000001 MOV #BIT0,R1 ;INITIALIZE THE SELECTED LINE BIT MASK.
4005 017040 016702 163176 MOV TXAD2A,R2 ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
4006 017044 005202 INC R2 ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
4007 017046 012703 000020 MOV #NUMLNS,R3 ;GET MAXIMUM LINE NUMBER PLUS ONE.
4008 017052 016704 163200 MOV IESTAT,R4 ;GET THE STATES OF THE INT ENABLE BITS.
4009 017056 005005 CLR R5 ;LOG POSSIBLE TX DISABLED ON ALL LINES.
4010
4011 ;* SELECT EVERY LINE IN TURN, AND LOG THE STATE OF EACH TX.ENABLE BIT.
4012 017060 010477 163142 2$: MOV R4,#CSRA ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
4013 017064 105712 TSTB (R2) ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
4014 017066 100001 BPL 4$ ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
4015 017070 050105 BIS R1,R5 ;LOG TX ENABLE BIT SET FOR SELECTED LINE.
4016
4017 ;* CLEAR TX.ENABLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX DISABLE
4018 ;* LINE BIT MAP.
4019
4020 017072 030100 4$: BIT R1,R0 ;CHECK STATE OF DISABLE LINE BIT MAP.
4021 017074 001402 BEQ 6$ ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
4022 017076 142712 000200 BICB #BIT7,(R2) ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
4023 017102 005204 6$: INC R4 ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
4024 017104 006301 ASL R1 ;SHIFT BIT MAP FOR NEXT LINE.
4025 017106 005303 DEC R3 ;DECREMENT LINE NUMBER.
4026 017110 001363 BNE 2$ ;LOOP TO CHECK NEXT LINE.
4027
4028 017112 010566 000014 60$: PASS R5 ;RESTORE GPRS,EXCEPT
4029 017116 004736 MOV R5,R5$LOT(SP) ;PUT R5 IN STACK SLOT.
4030 017120 000207 JSR PC,#(SP) ;RETURN TO PREG05 SUBRT.
;R5 - PREVIOUS STATES OF ALL TX.ENABLE BITS.
RTS PC

```

```

4032 .SBTTL GLOBAL SUBROUTINE - TXENBL -
4033 ;* *****
4034 ;* - TRANSMITTER ENABLE -
4035 ;* THIS SUBROUTINE IS USED TO ENABLE TRANSMISSION ON SELECTED LINES BY
4036 ;* SETTING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
4037 ;*
4038 ;* INPUTS: R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO SET TX.ENABLE.
4039 ;* CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
4040 ;* IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
4041 ;* NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
4042 ;* TXAD2A - CONTAINS THE ADDRESS OF THE TBUFAD2 REGISTER.
4043 ;*
4044 ;* OUTPUTS: R5 - BIT'S SET INDICATE PREVIOUSLY DISABLED LINES.
4045 ;* TBUFAD2 - THE STATE OF THE TX.ENABLE BIT MAY BE ALTERED.
4046 ;* THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
4047 ;*
4048 ;* CALLING SEQUENCE: JSR PC,TXENBL
4049 ;*
4050 ;* COMMENTS:
4051 ;*
4052 ;* SUBORDINATE ROUTINES CALLED: NONE.
4053 ;* *****
4054
4055 TXENBL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
; R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4056 017122 004567 164732 JSR
MOV R5,R0 ;COPY BIT MAP OF LINES TO ENABLE.
4057 017126 010500 MOV #BIT0,R1 ;INITIALIZE THE SELECTED LINE BIT MASK.
4058 017130 012701 000001 MOV TXAD2A,R2 ;GET THE ADDRESS OF THE TBUFAD2 REGISTER.
4059 017134 016702 163102 MOV TXAD2A,R2 ;GET THE ADDRESS OF THE MSBYTE OF TBUFAD2 REG.
4060 017140 005202 INC R2 ;GET MAXIMUM LINE NUMBER.
4061 017142 012703 000020 MOV #NUMLNS,R3 ;GET THE STATES OF THE INT ENABLE BITS.
4062 017146 016704 163104 MOV IESTAT,R4 ;CLEAR TX.ENABLE BIT LOG OF DISABLED LINES.
4063 017152 005005 CLR R5
4064 ;*
4065 ;* SELECT EVERY LINE IN TURN,AND LOG ANY TX.ENABLE BIT THAT IS CLEAR.
4066 017154 010477 163046 2$: MOV R4,CSRA ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
4067 017160 105712 TSTB (R2) ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
4068 017162 100401 BMI 4$ ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
4069 017164 050105 BIS R1,R5 ;LOG TX ENABLE BIT CLEAR FOR SELECTED LINE.
4070 ;*
4071 ;* SET TX.ENABLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX ENABLE
4072 ;* LINE BIT MAP.
4073 4$: BIT R1,R0 ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
4074 017166 030100 BEQ 6$ ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
4075 017170 001402 BISB #BIT7,(R2) ;ENABLE TRANSMISSION ON SELECTED LINE.
4076 017172 152712 000200 INC R4 ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
4077 017176 005204 ASL R1 ;SHIFT BIT MAP FOR NEXT LINE.
4078 017200 006301 DEC R3 ;DECREMENT LINE NUMBER.
4079 017202 005303 BNE 2$ ;LOOP TO CHECK NEXT LINE.
4080 017204 001363
4081 60$: PASS R5 ;RESTORE GPRS,EXCEPT
4082 017206 010566 000014 MOV R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
4083 017212 004736 JSR PC,@(SP) ;RETURN TO PREG05 SUBRT.
4084 ; R5 - LINE BIT MAP CORRESPONDING TO THE
4085 017214 000207 RTS PC ; PREVIOUS LINES THAT WERE DISABLED.

```

CZDHYBO DHU-11 FUNC TST PART2
GLOBAL SUBROUTINE

MACRO M1200 15-MAR-84 09:28 PAGE 87
- TXIEO -

SEQ 112

```

4087      .SBTTL GLOBAL SUBROUTINE      . TXIEO -
4088      ;** *****
4089      ;*      - TRANSMITTER INTERRUPT DISABLE -
4090      ;*      THIS ROUTINE IS USED TO DISABLE TRANSMITTER INTERRUPTS IN THE DHU11.
4091      ;*
4092      ;* INPUTS:      NONE.
4093      ;*
4094      ;* OUTPUTS:     THE TX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
4095      ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
4096      ;*               ENABLE BITS.
4097      ;*
4098      ;* CALLING SEQUENCE:  JSR      PC,TXIEO
4099      ;*
4100      ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
4101      ;*               THE DUT CSR ARE DESTROYED.
4102      ;*
4103      ;* SUBORDINATE ROUTINES CALLED: NONE.
4104      ;* *****
4105 017216 010046 TXIEO:: MOV      RO,-(SP)      ;SAVE CONTENTS OF RO ON THE STACK.
4106 017220      GETPRI  -(SP)      ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
4107 017224      SETPRI  @PRI07      ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
4108 017232 042767 177677 163016      BIC      @177677,IESTAT ;CLEAR TX.INT.ENBL BIT IN IESTAT.
4109 017240 016777 163012 162760      MOV      IESTAT,@CSRA ;DISABLE TX INTERRUPTS.
4110 017246      SETPRI (SP)+      ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
4111 017252 012600      MOV      (SP)+,RO ;RESTORE RO.
4112 017254 000207      RTS      PC

```

```

4114      .SBTTL  GLOBAL SUBROUTINE          - TXIE1 -
4115      ;* *****
4116      ;*          - TRANSMITTER INTERRUPT ENABLE -
4117      ;*          THIS ROUTINE IS USED TO ENABLE TRANSMITTER INTERRUPTS IN THE DHU11.
4118      ;*
4119      ;* INPUTS:      NONE.
4120      ;*
4121      ;* OUTPUTS:     THE TX.INT.ENBL BIT IS SET IN THE DUT CSR.
4122      ;*               IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
4123      ;*               ENABLE BITS.
4124      ;*
4125      ;* CALLING SEQUENCE:  JSR      PC,TXIE1
4126      ;*
4127      ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
4128      ;*               THE DUT CSR ARE DESTROYED.
4129      ;*
4130      ;* SUBORDINATE ROUTINES CALLED: NONE.
4131      ;* *****
4132
4133 017256 052767 040000 162772 TXIE1:: BIS    0BIT14,IESTAT  ;SET TX.INT.ENBL BIT IN IESTAT.
4134 017264 042767 137677 162764      BIC    0137677,IESTAT  ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
4135 017272 016777 162760 162726      MOV    IESTAT,0CSRA   ;ENABLE TX INTERRUPTS.
4136 017300 000207          RTS    PC

```

```

4138 .SBTTL GLOBAL SUBROUTINE - UNSDIV -
4139 ;* *****
4140 ;* - UNSIGNED DIVIDE ROUTINE -
4141 ;* THIS SUBROUTINE IS USED TO DIVIDE A 32 BIT UNSIGNED DIVIDEND BY A
4142 ;* 16 BIT UNSIGNED DIVISOR GIVING A 16 BIT QUOTIENT. ALL NUMBERS ARE
4143 ;* CONSIDERED TO BE UNSIGNED. A SUCCESS FLAG IS NOT SET ON RETURN IF
4144 ;* THE QUOTIENT WAS TOO BIG TO BE CONTAINED IN 16 BITS.
4145 ;*
4146 ;* INPUTS: R1 - THE DIVISOR, UNSIGNED, 16 BITS.
4147 ;* R2 - MOST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
4148 ;* R3 - LEAST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
4149 ;*
4150 ;* OUTPUTS: R1 - QUOTIENT, UNSIGNED, 16 BITS (177777 IF OVERFLOW).
4151 ;* CARRY - SUCCESS FLAG, SET IF COMPLETE QUOTIENT FITS IN 16 BITS.
4152 ;*
4153 ;* CALLING SEQUENCE: JSR PC,UNSDIV
4154 ;*
4155 ;* COMMENTS: IF THE DIVISOR IS 0 THE QUOTIENT IS RETURNED AS ALL ONES
4156 ;* (177777) AND THE CARRY IS CLEAR REGARDLESS OF THE DIVIDEND.
4157 ;*
4158 ;* SUBORDINATE ROUTINES CALLED: NONE.
4159 ;* - *****
4160
4161 017302 UNSDIV:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
4162 017302 004567 164552 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4163
4164 ;* CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
4165 ;*
4166 017306 010204 MOV R2,R4 ;GET MSW OF DIVIDEND FOR SUBTRACT.
4167 017310 160104 SUB R1,R4 ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
4168 017312 103403 BCS 2$ ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
4169 017314 012701 177777 MOV 0-1,R1 ;SET QUOTIENT TO ALL ONES (177777),
4170 017320 000442 BR 60$ ;EXIT WITH CARRY CLEAR.
4171
4172 ;* SET UP COUNTERS AND VARIOUS WORKING GPRS.
4173 017322 005004 2$: CLR R4 ;CLEAR THE LSW OF THE DIVISOR.
4174 017324 000241 CLC ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
4175 017326 006001 ROR R1 ;DIVISOR BY
4176 017330 006004 ROR R4 ;2(UNSIGNED)
4177 017332 012700 000020 MOV 016.,R0 ;SET UP INITIAL SHIFT COUNT TO 16.
4178
4179 ;* THE SUBTRACT AND SHIFT LOOP.
4180 ;*
4181 017336 010246 4$: MOV R2,-(SP) ;SAVE MSWORD OF DIVIDEND.
4182 017340 010346 MOV R3,-(SP) ;SAVE LSWORD OF DIVIDEND.
4183 017342 160403 SUB R4,R3 ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
4184 017344 005602 SBC R2 ;MSWORD DIVIDEND - BORROW.
4185 017346 103402 BCS 6$ ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
4186 017350 160102 SUB R1,R2 ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
4187 017352 103003 BCC 8$ ;IF NO BORROW, IT WENT, CARRY IS CLEAR.
4188 ;*
4189 ;* IT DIDN'T GO, SO WE SHIFT A 1 INTO THE QUOTIENT (COMPLEMENTED LATER).
4190 ;* CARRY IS SET.
4191 ;*
4192 017354 012603 6$: MOV (SP)+,R3 ;RESTORE LSWORD OF DIVIDEND.
4193 017356 012602 MOV (SP)+,R2 ;RESTORE MSWORD OF DIVIDEND.

```

```

4194 017360 000401      BR      10$      ;GOTO SHIFT 1 INTO THE QUOTIENT.
4195
4196      ;+
4197      ; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
4198      ; COMPLEMENTED LATER). CARRY IS CLEAR.
4199 017362 012626      8$:  MOV      (SP)+,(SP)+      ;POP THE SAVED DIVIDEND OFF OF THE STACK.
4200
4201      ;+
4202      ; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.
4203 017364 006105      10$:  ROL      R5      ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
4204 017366 000241      CLC      ;DIVIDE THE
4205 017370 006001      ROR      R1      ; DIVISOR BY
4206 017372 006004      ROR      R4      ; 2 (UNSIGNED).
4207 017374 005300      DEC      R0      ;COUNT THIS SHIFT AND SUBTRACT.
4208 017376 001357      BNE      4$      ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
4209 017400 005105      COM      R5      ;GET QUOTIENT FROM INVERTED QUOTIENT.
4210
4211      ;+
4212      ; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.
4213 017402 000241      CLC      ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
4214 017404 006103      ROL      R3      ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
4215 017406 103402      BCS      12$      ;IF CARRY FROM SHIFT, ROUND UP.
4216 017410 160403      SUB      R4,R3      ;SUBTRACT DIVISOR FROM DIVIDEND.
4217 017412 103403      BCS      14$      ;IF BORROW, DON'T ROUND UP.
4218
4219      ;+
4220      ; ROUND UP, EXTRA SUBTRACT WENT.
4221 017414 005205      12$:  INC      R5      ;INCREMENT THE QUOTIENT BY ONE.
4222 017416 001001      BNE      14$      ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
4223 017420 005305      DEC      R5      ;DON'T LET ROUNDING CAUSE OVERFLOW.
4224
4225      ;+
4226      ; ALL DONE, PASS QUOTIENT AND EXIT.
4227 017422 010501      14$:  MOV      R5,R1      ;PASS QUOTIENT BACK IN R1.
4228 017424 000261      SEC      ;INDICATE NO OVERFLOW.
4229
4230 017426      60$:  PASS      R1      ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
      017426 010166 000004      MOV      R1,R1$LOT(SP)      ;PUT R1 IN STACK SLOT.
      017432 004736      JSR      PC,$(SP)+      ;RETURN TO PREGO'S SUBRT.
4231
4232 017434 000207      RTS      PC      ;R1 - 16 BIT, UNSIGNED QUOTIENT.
      ;CARRY - SET INDICATES NO OVERFLOW (SUCCESS).

```

```

4234      ,SBTTL  GLOBAL SUBROUTINE                - WAIBIS -
4235      ; * *****
4236      ; * - WAIT FOR BIT SET ROUTINE -
4237      ; * THIS SUBROUTINE WAITS FOR THE SPECIFIED BIT TO BECOME SET.  IF THE
4238      ; * SPECIFIED BIT GOES TO A SET STATE WITHIN THE SPECIFIED TIME-OUT
4239      ; * PERIOD A SUCCESS INDICATION IS RETURNED BY THIS ROUTINE.
4240      ; * THE LAST VALUE WHICH IS READ LOOKING FOR THE CONDITION IS RETURNED TO
4241      ; * ALLOW THE USE OF THIS ROUTINE TO LOOK FOR DESTRUCTIVE READ CONDITIONS.
4242      ; *
4243      ; * INPUTS:      R1 - TIME-OUT VALUE AND BIT NUMBER INDICATION:
4244      ; *                BITS 15 THRU 12 - NUMBER OF BIT TO TEST (RANGE 0 THRU 15).
4245      ; *                BITS 11 THRU 0 - TIME-OUT VALUE IN MILLI-SECONDS (4095 MAX).
4246      ; *                R2 - ADDRESS OF WORD CONTAINING THE BIT TO TEST.
4247      ; *                MSLCNT.
4248      ; *
4249      ; * OUTPUTS:     R2 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
4250      ; *                CARRY - SUCCESS FLAG (CARRY SET IF BIT SET BEFORE TIME-OUT).
4251      ; *
4252      ; * CALLING SEQUENCE:  MOV      #130040,R1      ;PASS BIT 11 (13 OCTAL) AND
4253      ; *                                ; 32 (40 OCTAL) MS DELAY.
4254      ; *                                MOV      #LABEL,R2      ;TEST BIT IN WORD AT "LABEL".
4255      ; *                                JSR      PC,WAIBIS      ;WAIT 32 MS FOR BIT 11 TO SET.
4256      ; *
4257      ; * COMMENTS:
4258      ; *
4259      ; * SUBORDINATE ROUTINES CALLED: MSLGET.
4260      ; * *****
4261
4262      017436      WAIBIS:: SAVE                      ;SAVE CONTENTS OF GPRS R0 THRU R5.
4263      017436      004567      164416      JSR      R5,PREG05      ;CALL REGISTER SAVE SUBRT.
4264      017442      010204      MOV      R2,R4      ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
4265      017444      010102      MOV      R1,R2
4266      017446      042701      170000      BIC      #170000,R1      ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
4267      017452      042702      007777      BIC      #7777,R2      ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
4268      017456      000302      SWAB     R2      ;PUT LINE NUMBER FIELD IN LS BYTE.
4269      017460      006202      ASR      R2      ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
4270      017462      006202      ASR      R2      ; POSITION TO USE IT AS A WORD TABLE OFFSET
4271      017464      006202      ASR      R2      ; FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
4272      017466      016202      002336      MOV      BITTBL(R2),R2      ;GET BIT MAP OF LINE TO TEST FROM TABLE.
4273      017472      010203      MOV      R2,R3      ;INDICATE THAT THE BIT SHOULD BE SET.
4274      017474      004767      175230      JSR      PC,MSLGET      ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
4275      017500      010002      MOV      R0,R2      ; CARRY IS CORRECT UPON MSLGET RETURN.
4276      017502      010266      000006      60$:    PASS     R2      ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
4277      017506      004736      MOV      R2,R2SLOT(SP)      ;RESTORE GPRS, EXCEPT THE FOLLOWING:
4278      017510      000207      JSR      PC,@(SP)      ;PUT R2 IN STACK SLOT.
4279      ; R2 - LAST VALUE READ LOOKING FOR CONDITION.
4280      ; CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).

```

```

4280      ,SBTTL GLOBAL SUBROUTINE - WAITTX -
4281      ;+ *****
4282      ;* - WAIT FOR TX TO FINISH -
4283      ;* THIS SUBROUTINE IS USED IN THE FIHAVL.TST.
4284      ;* IT WAITS FOR TRANSMISSION TO COMPLETE IE TX_ACTION. THEN DELAYS
4285      ;* FOR 5 MILLISECONDS TO ALLOW TIME FOR THE LAST CHARACTER TO GET INTO
4286      ;* THE FIFO.
4287      ;*
4288      ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR.
4289      ;*
4290      ;* OUTPUTS: CARRY - SET INDICATES SUCCESS.
4291      ;*
4292      ;* CALLING SEQUENCE: JSR PC,WAITTX
4293      ;*
4294      ;* COMMENTS:
4295      ;*
4296      ;* SUBORDINATE ROUTINES CALLED: DELAY,WAIBIS.
4297      ;+ *****
4298
4299      WAITTX:: SAVE
4300      017512 004567 164342      JSR      R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
4301      017516 012701 170536      MOV      #170536,R1 ;CALL REGISTER SAVE SUBRT.
4302      017522 016702 162500      MOV      CSRA,R2 ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
4303      017526 004767 177704      JSR      PC,WAIBIS ;PASS THE ADDRESS OF THE CSR.
4304      017532 103005      BCC      60$ ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
4305      017534 012704 000005      MOV      #5,R4 ;BRANCH IF NO TX_ACTION, ABORT THE TEST.
4306      017540 004767 174516      JSR      PC,DELAY ;PASS DELAY OF 5 MILLI SECS.
4307      017544 000261      SEC ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
4308      017546      60$: PASS ;SET CARRY TO INDICATE SUCCESS.
4309      017546 004736      JSR      PC,@(SP)+ ;RESTORE GPRS.
4310      017550 000207      RTS      PC ;RETURN TO PREG05 SUBRT.

```

B10

CZDHVBO DHU-11 FUNC TST PART2
GLOBAL SUBROUTINE

MACRO M1200 15-MAR-84 09:28 PAGE 92
- WTWLNC -

SEQ 118

```

4312 .SBTTL GLOBAL SUBROUTINE - WTWLNC
4313 *** *****
4314 ;* - LINE CONTROL REGISTER SETUP ROUTINE -
4315 ;* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
4316 ;* CONTROL REGISTERS (LNCTRL) TO THE SPECIFIED STATE. ONLY THE LNCTRLS
4317 ;* FOR THE SPECIFIED LINES ARE ALTERED.
4318 ;*
4319 ;* INPUTS: R0 - NEW LINE PARAMETERS.
4320 ;* R5 - BIT MAP OF LINES TO BE ALTERED.
4321 ;* CSRA - CONTAINS ADDRESS OF THE DUT CSR.
4322 ;* IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
4323 ;* ENABLE BITS IN THE CSR.
4324 ;* LNCTRA - CONTAINS ADDRESS OF THE DUT LNCTRL REGISTERS.
4325 ;*
4326 ;* OUTPUTS: LNCTRL - SPECIFIED DUT LINE CONTROL REGISTERS ARE ALTERED.
4327 ;*
4328 ;* CALLING SEQUENCE: JSR PC,WTWLNC
4329 ;*
4330 ;* COMMENTS:
4331 ;*
4332 ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
4333 ;* - *****
4334
4335 017552 WTWLNC:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
017552 004567 164302 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4336 ;*
4337 ;* SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
4338 ;*
4339 017556 016701 162454 MOV LNCTRA,R1 ;SET UP THE REGISTER ADDRESS PARAMETER.
4340 017562 010002 MOV R0,R2 ;SET UP THE DESIRED REGISTER CONTENTS.
4341 017564 010503 MOV R5,R3 ;SET UP THE BIT MAP OF LINES TO ALTER.
4342 017566 012704 177777 MOV 0-1,R4 ;SELECT ALL REGISTER BITS TO BE ALTERED.
4343 ;*
4344 ;* CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
4345 ;*
4346 017572 004767 173572 JSR PC,ALTFLD ;ALTER THE REGISTER CONTENTS.
4347 ;*
4348 017576 004736 601: PASS ;RESTORE GPRS.
017576 004736 JSR PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
4349 017600 000207 RTS PC

```

```

4351 .SBTTL GLOBAL SUBROUTINE - WTWLPR -
4352 ;* *****
4353 ;* - LINE PARAMETER REGISTER SETUP ROUTINE -
4354 ;* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
4355 ;* PARAMETER REGISTERS (LPR) TO THE SPECIFIED STATE. ONLY THE LPRS FOR
4356 ;* THE SPECIFIED LINES ARE ALTERED.
4357 ;*
4358 ;* INPUTS: R0 - NEW LINE PARAMETERS.
4359 ;* R5 - BIT MAP OF LINES TO BE ALTERED.
4360 ;* CSRA - CONTAINS ADDRESS OF THE DUT CSR.
4361 ;* IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
4362 ;* ENABLE BITS IN THE CSR.
4363 ;* LPRA - CONTAINS ADDRESS OF THE DUT LPR.
4364 ;*
4365 ;* OUTPUTS: LPR - SPECIFIED DUT LINE PARAMTER REGISTERS ARE ALTERED.
4366 ;*
4367 ;* CALLING SEQUENCE: JSR PC,WTWLPR
4368 ;*
4369 ;* COMMENTS:
4370 ;*
4371 ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
4372 ;* *****
4373
4374 017602 WTWLPR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
017602 004567 164252 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4375 ;*
4376 ;* SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
4377 ;*
4378 017606 016701 162420 MOV LPRA,R1 ;SET UP THE REGISTER ADDRESS PARAMETER.
4379 017612 010002 MOV R0,R2 ;SET UP THE DESIRED REGISTER CONTENTS.
4380 017614 010503 MOV R5,R3 ;SET UP THE BIT MAP OF LINES TO ALTFLD.
4381 017616 012704 177777 MOV #-1,R4 ;SELECT ALL REGISTER BITS TO BE ALTERED.
4382 ;*
4383 ;* CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
4384 ;*
4385 017622 004767 173542 JSR PC,ALTFLD ;ALTER THE REGISTER CONTENTS.
4386
4387 017626 PASS ;RESTORE GPRS.
017626 004736 JSR PC,@(SP), ;RETURN TO PREG05 SUBRT.
4388 017630 000207 RTS PC

```

```

4390      .SBTTL  INTERRUPT SERVICE ROUTINE      - CLKINT
4391      ;** *****
4392      ;*      THIS ROUTINE IS EXECUTED CLKHRZ TIMES PER SECOND.  IT DECREASES THE
4393      ;*      TWO TIMER COUNTERS DOWN TO ZERO.
4394      ;*
4395      ;* INPUTS:      TIMER1 - TIMER COUNTER #1.
4396      ;*              TIMER2 - TIMER COUNTER #2.
4397      ;*              TIMER3 - TIMER COUNTER FOR CALL OF BREAK MACRO.
4398      ;*
4399      ;* OUTPUTS:     THE 2 TIMER COUNTERS ARE DECREMENTED IF THEY ARE NOT ZERO.
4400      ;*
4401      ;* CALLING SEQUENCE:  PUT #CLKINT IN THE CLOCK INTERRUPT VECTOR SLOT.
4402      ;*                   PUT THE DESIRED TIME PERIOD (SECONDS TIMES CLKHRZ) IN
4403      ;*                   EITHER TIMER1 OR TIMER2 AND POLL THE RESPECTIVE TIMER
4404      ;*                   COUNTER TO DETECT ITS GOING TO 0 ON TIME-OUT.
4405      ;*
4406      ;* COMMENTS:     THE 2 COUNTERS WILL NOT WRAPAROUND BUT WILL STOP AT 0.  THIS
4407      ;*                   ALLOWS THE DETECTION OF A TIME-OUT ANY TIME AFTER THE TIME-OUT
4408      ;*                   HAS OCCURRED UNTIL THE TIMER COUNTER IS SET TO ANOTHER VALUE.
4409      ;*
4410      ;* SUBORDINATE ROUTINES CALLED: NONE.
4411      ;-- *****
4412
4413 017632 005767 162456  CLKINT:: TST      TIMER1      ;CHECK FOR TIMER1 AT ZERO.
4414 017636 001402      BEQ      2$          ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
4415 017640 005367 162450      DEC      TIMER1      ;DECREMENT TIME COUNT.
4416 017644 005767 162446  2$:  TST      TIMER2      ;CHECK FOR TIMER2 AT ZERO.
4417 017650 001402      BEQ      4$          ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
4418 017652 005367 162440      DEC      TIMER2      ;DECREMENT TIME COUNT.
4419 017656 005367 162436  4$:  DEC      TIMER3      ;DECREMENT THE BREAK COUNT.
4420 017662 001006      BNE      60$          ;EXIT IF NOT TIME TO CALL BREAK.
4421 017664 016767 162432 162426  MOV      BCOUNT,TIMER3 ;SET UP TIME TILL NEXT BREAK.
4422 017672 010046      MOV      RO,-(SP)      ;SAVE CONTENTS OF RO FROM BREAK MACRO.
4423 017674      BREAK      ;CHECK FOR OPERATOR CONTROL/C.
4424 017676 012600      MOV      (SP)+,RO      ;RESTORE CONTENTS OF RO.
4425 017700 000002 60$:  RTI

```

E10

CZDHV80 DHU-11 FUNC TST PART2
INTERRUPT SERVICE ROUTINE

MACRO M1200 15-MAR-84 09:28 PAGE 95
- RXDECT -

SEQ 121

```

4427      .SBTTL  INTERRUPT SERVICE ROUTINE      - RXDECT -
4428      ;* *****
4429      ;*      - RX INT DECTION ROUTINE -
4430      ;*      THIS ROUTINE DETECTS AN RX INTERRUPT BY SETTING THE RXINTC WORD TO 1.
4431      ;*      THIS ROUTINE IS USED IN THE RXTIMER TESTS.
4432      ;*
4433      ;* INPUTS:      RXINTC - STORGE FOR THE INTERRUPT COUNT.
4434      ;*
4435      ;* OUTPUTS:     RXINTC - SET TO 1.
4436      ;*
4437      ;* CALLING SEQUENCE:  PUT THE ADDRESS OF THE LABEL RXDECT IN THE VECTOR
4438      ;*                      LOCATION.
4439      ;*
4440      ;* COMMENTS:     THIS ROUTINE DOES NOT READ THE RXFIFO.
4441      ;*
4442      ;* SUBORDINATE ROUTINES CALLED: NONE.
4443      ;* -- *****
4444
4445 017702 012767 000001 162356 RXDECT:: MOV    #1,RXINTC      ;INDICATE THAT AN RX-INT HAS OCCURED.
4446 017710 000002                      RTI

```



```

4478 .SBTTL INTERRUPT SERVICE ROUTINE - TXAINT -
4479 ;** *****
4480 ;* - TRANSMIT ACTION INTERRUPT SERVICE ROUTINE -
4481 ;* THIS ROUTINE HANDLES A TX INTERRUPT BY COUNTING THE INTERRUPT,
4482 ;* SETTING A FLAG IF THE TX_ACTION BIT IS CLEAR, AND READING THE CSR
4483 ;* UNTIL THE TX_ACTION BIT CLEARS OR THE MAXIMUM READ COUNT IS EXCEEDED.
4484 ;*
4485 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR.
4486 ;* TXINTC - HOLDS THE COUNT OF THE NUMBER OF TX INTERRUPTS.
4487 ;* TXINTF - TX INTERRUPT FLAGS.
4488 ;*
4489 ;* OUTPUTS: TXINTC - CONTAINS THE UPDATED TX INTERRUPT COUNT.
4490 ;* TXINTF - TX INTERRUPT FLAGS (BIT 15 SET IF TX_ACTION CLEAR
4491 ;* BIT 14 SET IF MAX READ COUNT EXCEEDED)
4492 ;*
4493 ;* CALLING SEQUENCE: PUT THE ADDRESS OF THE LABEL TXAINT IN THE VECTOR
4494 ;* LOCATION.
4495 ;*
4496 ;* COMMENTS:
4497 ;*
4498 ;* SUBORDINATE ROUTINES CALLED: NONE
4499 ;*
4500 ;-- *****
4501
4502 TXAINT:: SAVE
4503 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4504 MOV TXINTC,R1 ;GET THE TX-INT COUNT.
4505 INC R1 ;INCREMENT THE COUNT.
4506 BVC 2$ ;BRANCH IF NO OVERFLOW OCCURED.
4507 DEC R1 ;RESET THE COUNT TO 177777.
4508 2$: MOV R1,TXINTC ;SAVE THE NEW COUNT.
4509 MOV TXINTF,R3 ;GET THE TX-INT FLAGS.
4510 TST @CSRA ;READ THE CSR.
4511 BMI 4$ ;AVOID SETTING THE ERROR FLAG IF
4512 ;THERE IS A TX_ACTION.
4513 BIS @BIT15,R3 ;SET THE FLAG.
4514 4$: MOV R3,TXINTF ;UPDATE THE TX-INT FLAGS.
4515 MOV @32.,R2 ;SET THE MAX TX_ACTION READ COUNT.
4516 ;*
4517 ; READ THE CSR UNTIL THE TX_ACTION FIFO IS EMPTY OR THE MAX READ COUNT
4518 ; IS EXCEEDED.
4519 ;--
4520 6$: TST @CSRA ;READ THE CSR.
4521 BPL 60$ ;EXIT IF TX_ACTION FIFO IS EMPTY.
4522 DEC R2 ;DECREMENT THE MAX READ COUNT.
4523 BNE 6$ ;BRANCH TO READ ANOTHER TX_ACTION IF MAX READ
4524 ;COUNT IS NOT EXCEEDED.
4525 BIS @BIT14,TXINTF ;SET THE "MAX TX_ACTION COUNT EXCEEDED" FLAG.
4526 60$: PASS
4527 JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
RTI

```

H10

CZDHV80 DHU-11 FUNC TST PART2
INTERRUPT SERVICE ROUTINE

MACRO M1200 15-MAR-84 09:28 PAGE 98
- TXAINT -

SEQ 124

4536

4537

4538

4539

4540

4541

4542

4543

4544

4545 020030

020030

4546

4547 020030

020030 000167

020032 000000

4548

4549

4550

4551 020034

020034

020034 104425

.SBTTL REPORT CODING SECTION

;;

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

;;

BGNRPT

L\$RPT::

EXIT RPT

.WORD J\$JMP
.WORD L10017-2-

.EVEN

ENDRPT

L10017:
TRAP C\$RPT

CZDHV80 DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 99
 PROTECTION TABLE

SEQ 125

```

4553          .SBTTL  PROTECTION TABLE
4561
4562
4563          ;**
4564          ; THIS TABLE IS USED BY THE RUNTIME SERVICES
4565          ; TO PROTECT THE LOAD MEDIA.
4566          ;**
4567
4568 020036          BGNPROT
4569 020036
4569
4570 020036 177777          -1          ;OFFSET INTO P-TABLE FOR CSR ADDRESS
4571 020040 177777          -1          ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
4572 020040 177777          -1          ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
4573
4574 020044          ENDPROT
4575

```

PROTECTION TABLE

```

4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611 020044
      020044
4612
4613 020044
      020044 012700 000040
      020050 104447
4614 020052
      020052 103416
4615
4616 020054
      020054 012700 000037
      020060 104447
4617 020062
      020062 103556
4618
4619 020064
      020064 012700 000035
      020070 104447
4620 020072
      020072 103555
4621
4622 020074
      020074 012700 000036
      020100 104447
4623 020102
      020102 103161
4624 020104 000167 000540
4625 020110
4626 020110
      020110 104433
4627
4628
4629
4630 020112
      020112 012700 000114
      020116 104462
      020120 010001
4631 020122 012167 162156
4632 020126 012167 162154
4633 020132 012167 162152
4634 020136 012167 162150
4635 020142 026727 162144 000062
4636 020150 001004

      .SBTTL INITIALIZE SECTION
      ;+
      ;*****
      ;* THIS SECTION CONTAINS THE CODE WHICH IS PERFORMED AT THE BEGINNING OF
      ;* EACH PASS OR AFTER A CONTINUE COMMAND.
      ;* THIS CODE PERFORMS THE FOLLOWING ACTIONS:
      ;*
      ;* MOVES THE INFORMATION HELD IN THE HARDWARE P-TABLE INTO THE GLOBAL
      ;* DATA AREA.
      ;*
      ;*****
      ;--
      BGNINIT
      L$INIT::
;SEE IF PROGRAM JUST STARTED, BR IF YES
      READEF $EF.START
      MOV $EF.START,RO
      TRAP C$REFG
      BCOMPLETE NEWSTA
      BCS NEWSTA
;SEE IF PROGRAM JUST RESTARTED, BR IF YES
      READEF $EF.RESTART
      MOV $EF.RESTART,RO
      TRAP C$REFG
      BCOMPLETE NEWRES
      BCS NEWRES
;SEE IF THIS IS A NEW PASS, BR IF YES
      READEF $EF.NEW
      MOV $EF.NEW,RO
      TRAP C$REFG
      BCOMPLETE NEWPAS
      BCS NEWPAS
;SEE IF PROGRAM WAS JUST CONTINUED
      READEF $EF.CONTINUE
      MOV $EF.CONTINUE,RO
      TRAP C$REFG
      BCOMPLETE GETPRM
      BCC GETPRM
      JMP ENDIT
NEWSTA:
      BRESET ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
      TRAP C$RESET
      ;+
      ; SET UP FOR LINE TIME CLOCK INTERRUPTS.
      ;--
      CLOCK L,R1 ;GET THE CLOCK PARAMETERS.
      MOV $'L,RO
      TRAP C$CLK
      MOV RO,R1
      MOV (R1)+,CLKCSR ;STORE CLOCK CSR ADDRESS.
      MOV (R1)+,CLKBRL ;STORE CLOCK BUS REQ INT LEVEL.
      MOV (R1)+,CLKVEC ;STORE CLOCK INTERRUPT VECTOR.
      MOV (R1)+,CLKHRZ ;STORE CLOCK FREQUENCY.
      CMP CLKHRZ,#50. ;TEST FOR 50HZ LINE FREQUENCY.
      BNE 2$ ;BRANCH IF CLOCK IS NOT 50HZ.

```

```

4637 020152 012767 000024 162144      MOV    #20.,MSTICK      ;INDICATE 20MS PER CLOCK TICK.
4638 020160 000403                BR      4$
4639 020162 012767 000021 162134 2$:  MOV    #17.,MSTICK      ;INDICATE 17 MS PER CLOCK TICK.
4640 020170 012746 000300                SETVEC  CLKVEC,#CLKINT,#PRI06 ;INITIALIZE CLOCK INTERRUPT VECTOR.
                                MOV    #PRI06,-(SP)
                                MOV    #CLKINT,-(SP)
                                MOV    CLKVEC,-(SP)
                                MOV    #3,-(SP)
                                TRAP    C$SVEC
                                ADD     #10,SP
4641 020216 016700 162070                MOV    CLKHRZ,RO      ;INITIALIZE THE BREAK COUNT
4642 020222 006300                ASL     RO              ; TO CAUSE A BREAK
4643 020224 010067 162072                MOV    RO,BCOUNT      ; EVERY 2 SECONDS.
4644 020230                SETPRI   #PRI05      ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
                                MOV     #PRI05,RO
                                TRAP    C$SPRI
4645 020234 104441
4645 ;+
4646 ; ENABLE THE LINE TIME CLOCK (LTC) CHECKING TO MAKE SURE THAT THE CSR
4647 ; IS ACCESSABLE.
4648 ; FIRST SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
4649 ;-
4650 020236 016767 157542 162032      MOV    4,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
4651 020244 012767 017712 157532      MOV    #TP4RTN,4    ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4652 ;+
4653 ; ENABLE LTC CHECKING FOR 004 TRAP IN CASE CSR IS NOT THERE.
4654 ;-
4655 020252 005067 162022                CLR     TP4FLG      ;CLEAR THE 004 TRAP FLAG.
4656 020256 012767 000100 162016      MOV    #BIT6,WORD1  ;SET UP TO SET BIT6 OF THE LTC CSR.
4657 020264 012700 002302                MOV    #WORD1,RO    ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
4658 020270 016701 162010                MOV    CLKCSR,R1    ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
4659 020274 004767 173666                JSR     PC,CKTRAP   ;MOVE AND CHECK FOR TRAP.
4660 020300 016767 161772 157476      MOV    TP4VEC,4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
4661 020306 103403                BCS     6$          ;IF NO TRAP, LTC IS THERE SO CONTINUE.
4662 020310 005067 161776                CLR     CLKHRZ      ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
4663 020314 000402                BR      8$          ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
4664 ;+
4665 ; CALIBRATE THE DELAY ROUTINE MILLI-SECOND DELAY COUNT VALUE.
4666 ;-
4667 020316 004767 173230 6$: JSR     PC,CALMSL
4668 ;+
4669 ; CHECK FOR MEMORY MANAGEMENT PRESENT ON THIS MACHINE.
4670 ; IF MEM MGT IS PRESENT, DISABLE IT.
4671 ;-
4672 020322 016767 157456 161746 8$:  MOV    4,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
4673 020330 012767 017712 157446      MOV    #TP4RTN,4    ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4674 020336 005067 161736                CLR     TP4FLG      ;CLEAR THE 004 TRAP FLAG.
4675 020342 005067 161734                CLR     WORD1      ;PREPARE TO CLEAR THE MEM MGT SRO REGISTER.
4676 020346 012700 002302                MOV    #WORD1,RO    ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
4677 020352 016701 161752                MOV    MMSRO,R1    ;SELECT MEM MGT SRO REGISTER AS DESTINATION.
4678 020356 005067 161750                CLR     MMPRES      ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
4679 020362 005067 161746                CLR     MMENAB      ;INDICATE MEM MGT IS NOT ENABLED.
4680 020366 004767 173574                JSR     PC,CKTRAP   ;CLEAR THE MEM MGT SRO REG AND CHECK FOR TRAP.
4681 020372 016767 161700 157404      MOV    TP4VEC,4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
4682 020400 103003                BCC     10$         ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
4683 020402 012767 000001 161722      MOV    #1,MMPRES    ;INDICATE THAT MEM MGT IS PRESENT.
4684 020410 005067 161650 10$: CLR     PASCNT      ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4685 020414 000167 000006                JMP     NEWPAS      ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.

```

```

4686
4687 020420      NEWRES: BRESET          ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
      020420      104433                      TRAP      C$RESET
4688 020422      005067      161636      CLR      PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4689 020426
4690 020426      012767      177777      161570      MOV      #-1,UNITN      ;RESET LOGICAL DEVICE TO -1
4691
4692      ; INCREMENT THE PASS COUNTER, CORRECT FOR ANY OVERFLOW.
4693      ; THIS COUNTER IS USED IN THE ROM VERSION TEST.
4694      ;
4695 020434      005267      161624      INC      PASCNT          ;INCREMENT THE PASS COUNTER.
4696 020440      001002      BNE      GETPRM      ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
4697 020442      005367      161616      DEC      PASCNT          ;SET PASS COUNT TO 177777 OCTAL.
4698
4699      ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
4700 020446      GETPRM:
4701 020446      005267      161552      INC      UNITN          ;INCREMENT LOGICAL DEVICE NUMBER
4702 020452      026767      161546      161332      CMP      UNITN,L$UNIT      ;SEE IF MAXIMUM UNIT NO. EXCEEDED
4703 020460      002362      BGE      NEWPAS      ;BR IF YES
4704
4705 020462      GPWARD      UNITN,R1          ;GET P-TABLE POINTER INTO R1
      020462      016700      161536      MOV      UNITN,R0
      020466      104442      TRAP      C$GPHRD
      020470      010001      MOV      R0,R1
4706 020472      BCOMPLETE      30$          ;BR IF DEVICE AVAILABLE
      020472      103401      BCS      30$
4707 020474      000764      BR      GETPRM      ;SKIP THIS DEVICE
4708
4709
4710      ;***** HARDWARE PARAMETER MOVING CODE *****
4711 020476      012167      161524      30$:      MOV      (R1)+,CSRA      ;STORE DHU-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
4712 020502      012102      MOV      (R1)+,R2      ;GET THE RX INTERRUPT VECTOR ADDRESS.
4713 020504      010267      161504      MOV      R2,RXVECA      ;STORE RX INT VECTOR ADDRESS.
4714 020510      062702      000004      ADD      #4,R2      ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
4715 020514      010267      161476      MOV      R2,TXVECA      ;STORE TX INT VECTOR ADDRESS.
4716 020520      012167      161474      MOV      (R1)+,ACTLNS      ;STORE DHU-11 ACTIVE LINE BIT MAP
4717 020524      111167      161472      MOV      (R1),LOPBACK      ;STORE DHU-11 LOOPBACK MODE
4718
4719      ;
4720      ; CALCULATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
4721      ; DEVICE REGISTER ADDRESS TABLE.
4722      ;
4722 020530      016701      161472      MOV      CSRA,R1      ;COPY CSR ADDRESS
4723 020534      005201      INC      R1      ;INCREMENT CSR ADDRESS
4724 020536      005201      INC      R1      ; COPY BY 2.
4725 020540      012703      000007      MOV      #7,R3      ;SET UP REGISTER COUNT
4726 020544      012702      002230      MOV      #RBUFA,R2      ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
4727 020550      010122      12$:      MOV      R1,(R2)+      ;STORE REGISTER ADDRESS IN TABLE
4728 020552      005201      INC      R1      ;INCREMENT REGISTER ADDRESS
4729 020554      005201      INC      R1      ; BY 2, FOR THE NEXT DEVICE REGISTER.
4730 020556      005303      DEC      R3      ;DECREMENT REGISTER COUNT
4731 020560      001373      BNE      12$      ;LOOP IF NOT DONE
4732
4733      ;
4734      ; INITIALISE THE BMP CODE QUEUE.
4735      ;
4736 020562      012700      002412      MOV      #BMPCQB,R0      ;GET THE START ADDRESS OF THE QUEUE.
4737 020566      012701      002612      MOV      #BMPCQE,R1      ;GET THE END ADDRESS OF THE QUEUE.

```

```

4738 020572 010067 161612      MOV    R0,BMPCQP      ;SET THE POINTER TO THE START OF THE QUEUE.
4739 020576 005020      14$: CLR    (R0)+      ;CLEAR OUT THE CONTENTS OF THE QUEUE.
4740 020600 020001      CMP    R0,R1      ;CHECK IF END OF QUEUE HAS BEEN REACHED.
4741 020602 103775      BLO    14$      ;LOOP IF NOT ALL DONE.
4742
4743      ;+
4744      ; REPORT THE UNIT NUMBER IF THE SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
4745      ; AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
4746 020604 032767 000020 161376      BIT    #BIT4,OPTION      ;CHECK IF THE QUESTION WAS ANSWERED YES.
4747 020612 001416      BEQ    16$      ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
4748 020614 026727 161172 000001      LMP    L$UNIT,#1      ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
4749 020622 003412      BLE    16$      ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
4750 020624      PRINTF    #MFUNIT,UNITN      ;REPORT UNIT NUMBER.
      MOV    UNITN,-(SP)
      MOV    #MFUNIT,-(SP)
      MOV    #2,-(SP)
      MOV    SP,R0
      TRAP    C$PNTF
      ADD    #6,SP
4751 020650      16$:
4752
4753 020650 005067 161374      ENDIT: CLR    CTRLCF      ;CLR THE CTRL-C TEST ABORT FLAG.
4754      ;+
4755      ; SET THE PROCESSOR PRIORITY TO DISABLE ALL INTERRUPTS.
4756      ;+
4757 020654      SETPRI    #PRI07      ;SET PROCESSOR PRIORITY TO 7.
      MOV    #PRIC7,R0
      TRAP    C$SPRI
      020654 012700 000340
      020660 104441
4758 020662      ENDINIT
      L10021:
      020662 104411      TRAP    C$INIT
4759
4760      000000      TNUM == 0      ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.

```

4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780

.SBTTL AUTODROP SECTION

; **
; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
; THE "ADR" FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
; SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
; DROPPED FROM TESTING.
; **4781 020664
020664

BGNAUTO

L\$AUTO::

4782
47894790 020664
020664
020664 104461

ENDAUTO

L10022: TRAP C\$AUTO

B11

CZDHV80 DHU 11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 102
AUTODROP SECTION

SEQ 131

```

4799
4800
4801      .SBTTL  CLEANUP CODING SECTION
4802
4803      ;;;
4804      ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4805      ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
4806      ;;;
4807
4808      020666      BGNCLN
4809      020666
4810      020666      005767      161356
4811      020672      001401
4812      020674      104433
4813      020676      24:
4814
4823
4824      020676      EXIT      CLN
4825      020676      104432
4826      020700      000002
4827
4837
4838      .EVEN
4839
4840      020702      ENDCLN
4841      020702
4842      020702      104412

```

```

L10023:
TRAP      C1CLEAN

```

```

;DID WE GET HERE BY CTRL-C FROM TEST?
;CTRL-C FROM TEST? NO, SKIP BUS RESET.
;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
TRAP      C1RESET

```

```

TRAP      C1EXIT
L10023:

```

C11.

CZDHVRO DHU-11 LUNC IS PART2 MACRO M1200 15-MAR-84 09:28 PAGE 103

SEQ 132

```

4849
4850
4851      .SBTTL  DROP UNIT SECTION
4852
4853      ***
4854      ; THE DROP UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4855      ; TO NO LONGER BE TESTED.
4856      ***
4857
4858 020704      BGNDU
4859 020704
4860 020704      PRINTF #DRC?,RO      ;REPORT UNIT THAT HAS BEEN DROPPED.
4861 020704      010046
4862 020706      012746      020730      MOV      RO,-(SP)
4863 020712      012746      000002      MOV      #DROP,-(SP)
4864 020716      010600      MOV      #2,-(SP)
4865 020720      104417      MOV      SP,RO
4866 020722      062706      000006      TRAP     C$PNTF
4867 020726      000427      ADD       #6,SP
4868
4869      BR      EDROP      ;BRANCH AROUND THE MESSAGE.
4870 020730      045      101      040      DROP:  .ASCIZ/NA UNIT#D6NA DROPPED FROM FURTHER TESTING.#N/
4871 020733      125      116      111
4872 020736      124      045      104
4873 020741      066      045      101
4874 020744      040      104      122
4875 020747      117      120      120
4876 020752      105      104      040
4877 020755      106      122      117
4878 020760      115      040      106
4879 020763      125      122      124
4880 020766      110      105      122
4881 020771      040      124      105
4882 020774      123      124      111
4883 020777      116      107      056
4884 021002      045      116      000
4885
4886      .EVEN
4887 021006      EDROP:
4888
4889      EXIT      DU
4890
4891      .WORD      J$JMP
4892      .WORD      L10024-2-.
4893
4894      ENDDU
4895
4896      L10024:
4897      TRAP      C$DU
4898 021012
4899 021012
4900 021012      104453

```

D11

CZDHVBO DHU 11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 104

SEQ 133

DROP UNIT SECTION

```

4879
4880
4881      .SBTTL  ADD UNIT SECTION
4882
4883      ;**
4884      ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
4885      ; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
4886      ; TO THE TEST CYCLE.
4887      ;--
4888
4889      021014      BGNAU
4890      021014
4891
4892      021014      EXIT      AU
4893      021014      000167
4894      021016      000000
4895
4896      021020
4897      021020      104452
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

```

E11

CZDHYB0 DHU 11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 105
HARDWARE TEST - ADRA -

SEQ 134

```

4899 .SBTTL  HARDWARE TEST          - ADRA -
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 ;

021022 000001 161224
021022 000001 161224
021022 012767 000001 161224
021030 012767 177777 161212
021036 012767 000145 163006
021044 012767 005262 163002
021052 012767 011430 162776

021060 016767 156720 161210
021066 012767 017712 156710
021074 005005

021076 016700 161124
021102 012701 021274
021106 004767 173054
021112 103402
021114 052705 100001
021120 042767 000017 000146 4$
021126 010100
021130 016701 161072
021134 004767 173026
021140 103403
021142 052705 100002
021146 000434

021150 012702 000010
021154 016767 161046 000110
021162 016700 000104
021166 012701 021274
021172 004767 172770
021176 103402
021200 052705 100001
021204 010100
021206 016701 000060

BGNTST
T1::
TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
MOV @TNUM,TSTNUM ;SET UP THE TEST NUMBER. (1)
MOV @-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
MOV @101,ERRNBR ;SET THE TEST ERROR NUMBER IN THE TABLE.
MOV @EM0103,ERRMSG ;SET UP THE TEST FAILURE MESSAGE IN THE TABLE.
MOV @ER0101,ERRBLK ;SET-UP THE ERROR ROUTINE IN THE ERROR TABLE.

; SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
MOV 4,TP4VEC ;SAVE THE EXISTING 004 TRAP VECTOR.
MOV @TP4RTN,4 ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
CLR R5 ;CLEAR THE ERROR FLAGS.

; HERE BEGINS THE LOOP TO TEST THE REGISTERS FOR A LINE.
; FIRST TEST THE CSR AND SET THE IND.ADR.REG (I.A.R) FIELD.
MOV CSRA,R0 ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
MOV @52$,R1 ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
JSR PC,CKTRAP ;MOVE AND CHECK FOR TRAP.
BCS 4$ ;IF NO TRAP, BYPASS ERROR.
BIS @100001,R5 ;SET FATAL READ ERROR FLAGS.
BIC @17,52$ ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
MOV R1,R0 ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
MOV CSRA,R1 ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
JSR PC,CKTRAP ;MOVE AND CHECK FOR TRAP.
BCS 6$ ;IF NO TRAP, BYPASS ERROR.
BIS @100002,R5 ;SET FATAL WRITE ERROR FLAGS.
BR 40$ ;EXIT AND REPORT FATAL ERROR.

; NOW, WE TEST EACH REGISTER FOR THIS LINE.
6$: MOV @8$,R2 ;INIT REGISTER COUNTER TO 8.
MOV CSRA,50$ ;INITIALIZE THE REGISTER POINTER.
8$: MOV 50$,R0 ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
MOV @52$,R1 ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
JSR PC,CKTRAP ;PERFORM THE MOVE, CHECK FOR TRAP.
BCS 10$ ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
BIS @100001,R5 ;SET FATAL READ ERROR FLAGS.
10$: MOV R1,R0 ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
MOV 50$,R1 ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.

```

F11

CZDHVBO DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 105-1
HARDWARE TEST - ADRA -

SEQ 135

```

4955 021212 004767 172750      JSR    PC,CKTRAP      ;PERFORM THE MOVE, CHECK FOR TRAP.
4956 021216 103402              BCS    12$              ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
4957 021220 052705 100002      BIS    0100002,R5      ;SET FATAL WRITE ERROR FLAGS.
4958 021224 005267 000042      12$: INC    50$              ;INCREMENT THE REGISTER
4959 021230 005267 000036      INC    50$              ; POINTER BY 2.
4960 021234 005302              DEC    R2              ;COUNT THE REGISTER.
4961 021236 001351              BNE    8$              ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
4962
4963
4964
4965      ;+
4966      ; DONE CHECKING DEVICE REGISTER ADDRESSES.
4967      ; REPORT ANY ERRORS AND EXIT.
4968      ;-
4968 021240 016767 161032 156536 40$: MOV    TP4VEC,4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
4969 021246 005705              TST    R5              ;CHECK THE ERROR FLAGS.
4970 021250 100012              BPL    60$              ;EXIT ROUTINE IF NO ERRORS.
4971
4972      ;+
4973      ; REPORT "DEVICE REGISTER ACCESS TEST FAILED"
4974      ;-
4974 021252
4975 021252 104460
4976
4977 021254
4978 021254 016700 160744      DODU    UNITN      ;DROP THIS UNIT FROM FUTHER TESTING.
4979 021260 104451              MOV    UNITN,R0      ;
4978 021262 005067 160762      TRAP    C$DODU
4979 021266
4980 021266 104444              CLR    CTRLCF      ;INDICATE NO CTRL-C ABORT FROM TEST.
4981 021270 000402      DOCLN      ;ABORT THIS SUB PASS.
4982
4983 021270 000402      BR    60$              ;
4984
4985      ;***** LOCAL STORAGE. *****
4986      ;*****
4987 021272 000000      50$: .WORD    0      ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
4988 021274 000000      52$: .WORD    0      ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
4989
4990      ;***** END *****
4987 021276 005067 160746      60$: CLR    CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
4988 021302
4989 021302
4990 021302 104401      L10026: TRAP    C$ETST

```

```

4990 .SBTTL  HARDWARE TEST          - DMASTA -
4991 ;* *****
4992 ;*          - DMA START BIT TEST -
4993 ;*      THIS TEST VERIFIES THAT THE DMA_START BIT IN THE DUT'S LINE CONTROL
4994 ;*      REGISTERS WILL INITIATE DMA TRANSMISSION ON THE SELECTED LINE.
4995 ;*      THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, ON ALL ACTIVE LINES.
4996 ;*
4997 ;* *****
4998 021304      BGNTST
4999 021304          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.      T2::
021304          MOV      #PRI05,R0
021310          TRAP     C$SPRI
5000          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5001 021312 012767 000002 160734      MOV      #TNUM,ISTNUM          ;SET UP THE TEST NUMBER.      (40)
5002 021320 012767 177777 160722      MOV      #1,CTRLCF          ;INDICATE THAT WE ARE IN A TEST.
5003 021326 012767 000001 162514      MOV      #1,ERRTYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5004 021334 012767 007641 162510      MOV      #4001,ERRNBR          ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
5005 021342 012767 005403 162504      MOV      #EM4001,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERRTABL.
5006 021350 012767 013136 162500      MOV      #ER9101,ERRBLK          ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
5007 ;*
5008 ;* RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
5009 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
5010 ;* THIS SUBROUTINE REPORTS ERROR >>>> 4001 <<<<.
5011 ;*
5012 021356 004767 172634      JSR      PC,CLNRST          ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
5013 021362 103145      BCC      50$          ;RESET FAILURE?, ABORT THIS TEST.
5014 ;*
5015 021364 004767 173160      JSR      PC,INDATP          ;INITIALISE THE 256 BYTE DATA PATTERN.
5016 ;*
5017 ;* SET INTERNAL LOOPBACK,ENABLE RECEIVER FUNCTIONS ON ALL ACTIVE LINES.
5018 ;* SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
5019 ;* 2 STOP BITS.
5020 ;* ENABLE TRANSMITTERS ON ALL ACTIVE LINES.
5021 ;*
5022 021370 016705 160624      MOV      ACTLNS,R5          ;PASS THE ACTIVE LINE BIT MAP.
5023 021374 012700 000204      MOV      #204,R0          ;PASS THE LNCTRL CONTENTS.
5024 021400 004767 176146      JSR      PC,WTWLNLC          ;INITIALISE THE LNCTRL REGISTERS.
5025 021404 012700 177670      MOV      #177670,R0          ;PASS THE LPR CONTENTS.
5026 021410 004767 176166      JSR      PC,WTWLPR          ;INITIALISE THE LPR REGISTERS ON ALL LINES.
5027 021414 004767 175502      JSR      PC,TXENBL          ;ENABLE TRANSMITTERS ON ALL LINES.
5028 ;*
5029 ;* SET-UP OUTER LOOP TO TEST THE DMA_START BIT ON ALL ACTIVE LINES.
5030 ;*
5031 021420 016705 160574      MOV      ACTLNS,R5          ;GET THE ACTIVE LINE BIT MAP.
5032 021424 005001      CLR      R1          ;CLEAR THE LINE NUMBER COUNTER.
5033 021426 012767 007642 162416 2$:      MOV      #4002,ERRNBR          ;SET THE ERROR NUMBER TO 4002.
5034 021434 000241      CLC          ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
5035 021436 006005      ROR      R5          ;SHIFT THE BIT MAP INTO THE CARRY BIT.
5036 021440 103112      BCC      14$          ;DO NOT TEST THE LINE IF IT IS INACTIVE.
5037 021442 004767 173722      JSR      PC,PUFIFO          ;PURGE THE FIFO.
5038 021446 103113      BCC      50$          ;GO REPORT ERROR IF FIFO WILL NOT CLEAR.
5039 ;*
5040 ;* PERFORM DMA_START BIT TESTING ON EACH LINE INDIVIDUALLY.
5041 ;* TEST EACH DMA_START BIT BEFORE TX'ING DATA PATTERN, REPORT ERROR IF SET.
5042 ;* SET DMA_START BIT ON LUT, VERIFY IT IS SET, REPORT ERROR IF CLEAR.
5043 ;* WAIT FOR DMA TO COMPLETE.

```

```

5044                                     ; VERIFY DMA_START BIT IS CLEAR, REPORT ERROR IF SET.
5045                                     ; VERIFY CORRECT NUMBER OF CHARS WERE RECEIVED, REPORT ERROR IF < EXPECTED.
5046                                     ; -
5047 021450 005267 162376               INC     ERRNBR           ;SET ERROR NUMBER TO 4003.
5048 021454 012702 162650               MOV     @BUFBAS,R2       ;PASS THE START OF THE DATA PATTERN TO TX.
5049 021460 012703 000144               MOV     @1C0.,R3         ;PASS THE LENGTH OF THE DATA PATTERN.
5050 021464 004757 172710               JSR     PC,DODMA         ;TRANSMIT THE DATA PATTERN.
5051 021470 103067                       BCC     12$             ;GO REPORT ERROR IF DMA_START BIT SET.
5052                                     ; +
5053                                     ; TEST THE STATE OF THE DMA_START BIT ON THE LINE UNDER TEST.
5054                                     ; REPORT ERROR IF DMA_START BIT IS CLEAR.
5055                                     ; -
5056 021472 005267 162354               INC     ERRNBR           ;INCREMENT ERROR NUMBER TO 4004.
5057 021476 010177 160524               MOV     R1,@CSRA         ;SELECT THE LINE CURRENTLY UNDER TEST.
5058 021502 105777 160534               TSTB    @TXAD2A        ;TEST THE STATE OF THE DMA_START BIT.
5059 021506 100060                       BPL     12$             ;GO REPORT ERROR IF BIT IS CLEAR.
5060                                     ; +
5061                                     ; WAIT FOR DMA TRANSMISSION TO COMPLETE.
5062                                     ; -
5063 021510 005267 162336               INC     ERRNBR           ;INCREMENT ERROR NUMBER TO 4005.
5064 021514 010103                       MOV     R1,R3             ;SAVE THE LINE NUMBER.
5065 021516 012701 170226               MOV     @170226,R1       ;TEST BIT 15, TIMEOUT OF 150 MILLI SECS.
5066 021522 016702 160500               MOV     CSRA,R2         ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5067 021526 004767 175704               JSR     PC,WAIBIS        ;WAIT FOR DMA TO COMPLETE.
5068 021532 103045                       BCC     10$             ;GO REPORT ERROR IF TIMEOUT OCCURRED.
5069 021534 012704 000005               MOV     @5,R4           ;PASS DELAY OF 5 MILLI SECS.
5070 021540 004767 172516               JSR     PC,DELAY         ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5071 021544 010301                       MOV     R3,R1           ;RESTORE THE CURRENT LINE NUMBER.
5072                                     ; +
5073                                     ; TEST THE STATE OF THE DMA_START BIT ON THE LINE UNDER TEST.
5074                                     ; REPORT ERROR IF DMA_START BIT IS SET.
5075                                     ; -
5076 021546 005267 162300               INC     ERRNBR           ;INCREMENT ERROR NUMBER TO 4006.
5077 021552 010177 160450               MOV     R1,@CSRA         ;SELECT THE LINE CURRENTLY UNDER TEST.
5078 021556 105777 160460               TSTB    @TXAD2A        ;TEST THE STATE OF THE DMA_START BIT.
5079 021562 100432                       BMI     12$             ;GO REPORT ERROR IF BIT IS STILL SET.
5080                                     ; +
5081                                     ; VERIFY THE NUMBER OF CHARS RECEIVED = NUMBER OF CHARS EXPECTED.
5082                                     ; REPORT ERROR IF COUNT IS INCORRECT.
5083                                     ; IF MORE THAN 128 BMP CODES ARE FOUND THEN REPORT ERROR AND EXIT TEST.
5084                                     ; -
5085 021564 005003                       CLR     R3             ;CLEAR THE READ COUNTER.
5086 021566 012704 000200               MOV     @128.,R4        ;SET UP MAX BMP CODE READ COUNT.
5087 021572 012767 007647 162252 6$:   MOV     @4007.,ERRNBR   ;SET ERROR NUMBER TO 4007.
5088 021600 017702 160424               MOV     @RBUFA,R2       ;READ THE CHARACTER FROM THE FIFO.
5089 021604 100021                       BPL     12$             ;GO REPORT ERROR IF FIFO EMPTY TOO SOON.
5090 021606 012700 170301               MOV     @170301,R0      ;SET-UP BIT MASK OF A BMP CODE.
5091 021612 040200                       BIC     R2,R0           ;TRY TO CLEAR THE BMP CODE MASK.
5092 021614 001007                       BNE     8$             ;BRANCH IF NOT A BMP CODE.
5093 021616 005267 162230               INC     ERRNBR           ;INCREMENT ERROR NUMBER TO 4008.
5094 021622 004767 174652               JSR     PC,SAVBMP        ;SAVE THE BMP CODE ON THE QUEUE.
5095 021626 005304                       DEC     R4             ;DECREMENT MAX BMP CODE READ COUNT.
5096 021630 001422                       BEQ     50$             ;GO REPORT ERROR IF TOO MANY BMP CODES FOUND.
5097 021632 000757                       BR      6$             ;DO NOT COUNT THE BMP CODE AS A VALID CHAR.
5098 021634 005203                       INC     R3             ;COUNT THIS CHARACTER.
5099 021636 020327 000144               CMP     R3,@100.        ;HAVE WE RECIEVED 100 CHARACTERS?.
5100 021642 002753                       BLT     6$             ;LOOP UNTIL 100 (NON-BMP) CHARS ARE READ.

```

```

5101 021644 000410          BR      14$          ;SKIP AROUND THE ERROR REPORT.
5102
5103          ;*
5104          ; REPORT ERROR, SKIP FURTHER TESTING ON THIS LINE.
5105          ;*
5106 021646 010301          10$:  MOV     R3,R1          ;RESTORE THE CURRENT LINE NUMBER.
5107
5108 021650 012702 005435    12$:  MOV     $EM4002,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
5109                                ; "DMA_START BIT BAD ON LINE NN".
5110 021654                                ;          >>>>> ERROR <<<<<.
      021654 104460                                TRAP     C$ERROR
5111
5112          ;*
5113          ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5114          ;*
5115 021656 032767 000100 160324  BIT     $BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
5116 021664 001406          BEQ     60$          ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5117
5118 021666 005201          14$:  INC     R1          ;INCREMENT THE LINE NUMBER COUNTER.
5119 021670 005705          TST     R5          ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
5120 021672 001255          BNE     2$          ;YES; BRANCH TO TEST THE NEXT LINE.
5121 021674 000402          BR      60$          ;NO; EXIT THIS TEST.
5122
5123 021676 004767 174770    50$:  JSR     PC,TSABRT      ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5124 021702 005067 160342    60$:  CLR     CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5125
5126 021706          ENDTST
      021706
      021706 104401          L10027: TRAP     C$ETST

```

```

5128 .SBTTL  HARDWARE TEST          - DMABRT -
5129 ;+ *****
5130 ;+          - DMA ABORT/RESTART TEST -
5131 ;+ THIS TEST VERIFIES THAT EACH DMA_ABORT BIT WILL CORRECTLY HALT
5132 ;+ A DMA TRANSMISSION, AND RETURN A TX_ACTION.
5133 ;+ IT WILL ALSO VERIFY THAT THE ABORTED DMA TRANSMISSION CAN BE RESUMMED,
5134 ;+ AND THAT A TX_ACTION IS RETURNED UPON COMPLETION.
5135 ;+ THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, ON ALL ACTIVE LINES.
5136 ;+
5137 ;+ *****
5138 021710 BGNTST
5139 021710          T3::
          021710          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
          021710          012700 000240          MOV      #PRI05,R0
          021714          104441          TRAP     C$SPRI
5140          000003          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5141 021716 012767 000003 160330          MOV      #TNUM,TSTNUM          ;SET UP THE TEST NUMBER. (41)
5142 021724 012767 177777 160316          MOV      #-1,CTRLCF          ;INDICATE THAT WE ARE IN A TEST.
5143 021732 012767 000001 162110          MOV      #1,ERRTYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5144 021740 012767 010005 162104          MOV      #4101,ERRNBR          ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
5145 021746 012767 005471 162100          MOV      #EM4101,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERTBL.
5146 021754 012767 013136 162074          MOV      #ER9101,ERRBLK          ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
5147 ;+
5148 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
5149 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
5150 ; THIS SUBROUTINE REPORTS ERROR >>>> 4101 <<<<.
5151 ;+
5152 021762 004767 172230          JSR      PC,CLNRST          ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
5153 021766 103164          BCC      60$          ;RESET FAILURE?, ABORT THIS TEST.
5154 ;+
5155 021770 004767 172554          JSR      PC,INDATP          ;INITIALISE 256 BYTE DATA PATTERN.
5156 ;+
5157 ; SET INTERNAL LOOPBACK,ENABLE RECEIVER FUNCTIONS ON ALL ACTIVE LINES.
5158 ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
5159 ; 2 STOP BITS.
5160 ; ENABLE TRANSMITTERS ON ALL ACTIVE LINES.
5161 ;+
5162 021774 016705 160220          MOV      ACTLNS,R5          ;PASS THE ACTIVE LINE BIT MAP.
5163 022000 012700 000204          MOV      #204,R0          ;PASS THE LNCTRL CONTENTS.
5164 022004 004767 175342          JSR      PC,WTWLNLC          ;INITIALISE THE LNCTRL REGISTERS.
5165 022010 012700 175670          MOV      #177670,R0          ;PASS THE LPR CONTENTS.
5166 022014 004767 175562          JSR      PC,WTWLPR          ;INITIALISE THE LPR REGISTERS ON ALL LINES.
5167 022020 004767 175076          JSR      PC,TXENBL          ;ENABLE TRANSMITTERS ON ALL LINES.
5168 ;+
5169 ; PERFORM DMA_ABORT BIT TESTING ON EACH INDIVIDUAL (ACTIVE) LINE.
5170 ;+
5171 022024 016705 160170          MOV      ACTLNS,R5          ;GET THE ACTIVE LINE BIT MAP.
5172 022030 005001          CLR      R1          ;CLEAR THE LINE NUMBER COUNTER.
5173 022032 012767 010006 162012 2$:          MOV      #4102,ERRNBR          ;SET THE ERROR NUMBER TO 4102.
5174 022040 000241          CLC          ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
5175 022042 006005          ROR      R5          ;SHIFT THE BIT MAP INTO THE CARRY BIT.
5176 022044 103127          BCC      10$          ;DO NOT TEST THE LINE IF IT IS INACTIVE.
5177 022046 004767 173316          JSR      PC,PUFIFO          ;PURGE THE FIFO.
5178 022052 103130          BCC      50$          ;GO REPORT ERROR IF FIFO WILL NOT CLEAR.
5179 ;+
5180 ; CHECK THE DMA_ABORT BIT BEFORE ENABLING DMA, REPORT ERROR IF SET.
5181 ;+

```

5182	022054	005267	161772	
5183	022060	010177	160142	
5184	022064	032777	000001	160144
5185	022072	001105		
5186				
5187				
5188				
5189				
5190	022074	005267	161752	
5191	022100	012702	002650	
5192	022104	012703	000400	
5193	022110	004767	172264	
5194	022114	103107		
5195				
5196				
5197				
5198	022116	010177	160104	
5199	022122	012704	000050	
5200	022126	004767	172130	
5201	022132	052777	000001	160076
5202				
5203				
5204				
5205	022140	005267	161706	
5206	022144	010103		
5207	022146	012701	170012	
5208	022152	016702	160050	
5209	022156	004767	175254	
5210	022162	103050		
5211	022164	010301		
5212				
5213				
5214				
5215	022166	005267	161660	
5216	022172	012702	005557	
5217				
5218	022176	010177	160024	
5219	022202	105777	160034	
5220	022206	100441		
5221				
5222				
5223				
5224	022210	042777	000001	160020
5225	022216	052777	000200	160016
5226				
5227				
5228				
5229	022224	005267	161622	
5230	022230	010103		
5231	022232	012701	170536	
5232	022236	016702	157764	
5233	022242	004767	175170	
5234	022246	103016		
5235	022250	012704	000002	
5236	022254	004767	172002	
5237	022260	010301		
5238				

```

;+
INC ERRNBR ;INCREMENT ERROR NUMBER TO 4103.
MOV R1,0CSRA ;SELECT THE LINE CURRENTLY UNDER TEST.
BIT 0BIT0,0LNCTRA ;TEST THE STATE OF THE DMA_ABORT BIT.
BNE 6$ ;GO REPORT ERROR IF BIT IS SET.
;+
; ENABLE DMA TX ON SELECTED LINE, WAIT FOR DMA TO TX APPROX 1/4 OF DATA.
; ABORT THE DMA TRANSMISSION. WAIT FOR TX_ACTION TO BE RETURNED.
;-
INC ERRNBR ;SET ERROR NUMBER TO 4104.
MOV 0BUFBAS,R2 ;PASS THE START OF THE DATA PATTERN TO TX.
MOV 0256.,R3 ;PASS THE LENGTH OF THE DATA PATTERN.
JSR PC,DODMA ;TRANSMIT THE DATA PATTERN.
BCC 50$ ;GO REPORT ERROR IF THERE ARE TX PROBLEMS.
;+
; WAIT FOR DMA TO TRANSMIT 1/4 OF THE DATA BEFORE ABORTING.
;-
MOV R1,0CSRA ;SELECT THE LINE CURRENTLY UNDER TEST.
MOV 040.,R4 ;PASS THE DELAY TIME OF 40 MILLI SECONDS.
JSR PC,DELAY ;WAIT FOR APPROX 1/4 OF DATA TO BE TX'D.
BIS 0BIT0,0LNCTRA ;ABORT THE DMA TRANSMISSION.
;+
; WAIT FOR TX_ACTION TO BE RETURNED, REPORT ERROR IF TIME-OUT OCCURS.
;-
INC ERRNBR ;INCREMENT ERROR NUMBER TO 4105.
MOV R1,R3 ;SAVE THE LINE NUMBER.
MOV 0170012,R1 ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
MOV CSRA,R2 ;PASS THE ADDRESS OF THE REGISTER TO TEST.
JSR PC,WAIBIS ;WAIT FOR DMA TO COMPLETE.
BCC 4$ ;GO REPORT ERROR IF TIMEOUT OCCURRED.
MOV R3,R1 ;RESTORE THE CURRENT LINE NUMBER.
;+
; VERIFY DMA_START BIT CLEAR, REPORT ERROR IF SET.
;-
INC ERRNBR ;INCREMENT ERROR NUMBER TO 4106.
MOV 0EM4103,R2 ;SELECT MESSAGE TO BE REPORTED.
; "DMA_START BIT FOUND SET AFTER DMA ABORTED".
MOV R1,0CSRA ;SELECT THE LINE CURRENTLY UNDER TEST.
TSTB 0TXAD2A ;TEST THE STATE OF THE DMA_START BIT.
BMI 8$ ;GO REPORT ERROR IF IT IS SET.
;+
; RESUME DMA TRANSMISSION BY CLEARING DMA_ABORT AND SETTING DMA_START.
;-
BIC 0BIT0,0LNCTRA ;CLEAR THE DMA_ABORT BIT.
BIS 0BIT7,0TXAD2A ;SET THE DMA_START BIT.
;+
; WAIT FOR DMA TRANSMISSION TO COMPLETE.
;-
INC ERRNBR ;INCREMENT ERROR NUMBER TO 4107.
MOV R1,R3 ;SAVE THE LINE NUMBER.
MOV 0170536,R1 ;TEST BIT 15, TIMEOUT OF 350 MILLI SECS.
MOV CSRA,R2 ;PASS THE ADDRESS OF THE REGISTER TO TEST.
JSR PC,WAIBIS ;WAIT FOR DMA TO COMPLETE.
BCC 4$ ;GO REPORT ERROR IF TIMEOUT OCCURRED.
MOV 02,R4 ;PASS TIME-OUT OF 2 MILLI SECS.
JSR PC,DELAY ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
MOV R3,R1 ;RESTORE THE CURRENT LINE NUMBER.
;+

```

```

5239          ; TEST THE STATE OF THE DMA_ABORT BIT ON THE LINE UNDER TEST.
5240          ; REPORT ERROR IF DMA_ABORT BIT IS SET.
5241          ;-
5242 022262 005267 161564          INC   ERRNBR          ; INCREMENT ERROR NUMBER TO 4108.
5243 022266 010177 157734          MOV   R1, @CSRA        ; SELECT THE LINE CURRENTLY UNDER TEST.
5244 022272 032777 000001 157736  BIT   @BIT0, @LNCTRA    ; TEST THE STATE OF THE DMA_ABORT BIT.
5245 022300 001002                BNE   6$              ; GO REPORT ERROR IF BIT IS SET.
5246 022302 000410                BR    10$             ; BRANCH TO CHECK FOR ANY MORE LINES TO TEST.
5247          ;+
5248          ; REPORT ERROR, SKIP FURTHER TESTING ON THIS LINE.
5249          ;-
5250 022304 010301                4$:  MOV   R3, R1          ; RESTORE THE CURRENT LINE NUMBER.
5251
5252 022306 012702 005523                6$:  MOV   @EM4102, R2      ; PASS THE ERROR MESSAGE TO BE REPORTED.
5253                                ; "DMA_ABORT BIT BAD ON LINE NN".
5254 022312                                8$:  ERROR              ; >>>>> ERROR <<<<<.
5255                                TRAP   C$ERROR
5256                                022312 104460
5257
5258          ;+
5259          ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5260          ;-
5261 022314 032767 000100 157666          BIT   @BIT06, OPTION    ; EXIT WITH TEST FAILURE MESSAGE IF
5262 022322 001406                BEQ   60$              ; NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5263                                ; DURING THE SOFTWARE QUESTIONS.
5264          ;+
5265          ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
5266          ;-
5267 022324 005201                10$:  INC   R1              ; INCREMENT THE LINE NUMBER COUNTER.
5268 022326 005705                TST   R5              ; ARE THERE ANY MORE ACTIVE LINES TO TEST?.
5269 022330 001240                BNE   2$              ; YES; BRANCH TO TEST THE NEXT LINE.
5270 022332 000402                BR    60$             ; NO; EXIT THIS TEST.
5271 022334 004767 174332                50$:  JSR   PC, TSABRT      ; REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5272 022340 005067 157704                60$:  CLR   CTRLCF      ; INDICATE THAT WE ARE NOT WITHIN A TEST.
5273 022344                                ENDTST
5274 022344                                L10030:
5275 022344 104401                                TRAP   C$ETST

```

```

5275 .SBTTL  HARDWARE TEST - DMAERR -
5276 ;+ *****
5277 ;+ - DMA ERROR BIT TEST -
5278 ;+ THIS TEST VERIFIES THAT THE TX.DMA.ERROR BIT IN THE CSR IS
5279 ;+ FUNCTIONING CORRECTLY. THE DMA ERROR IS FORCED BY MAKING THE DUT
5280 ;+ ATTEMPT TO PERFORM A DMA TRANSFER FROM THE ADDRESS OF ITS OWN CSR.
5281 ;+ SINCE THE DEVICE CANNOT BE BOTH A BUS MASTER AND SLAVE AT THE SAME
5282 ;+ TIME, TIMEOUT WILL OCCUR WAITING FOR THE APPROPRIATE HANDSHAKE SIGNAL
5283 ;+ FROM THE DMA ADDRESS.
5284 ;+ THIS TEST IS PERFORMED IN INTERNAL LOOPBACK.
5285 ;+
5286 ;- *****
5287
5288 022346 BGNTST
5289 022346
5290 022346 012700 000240 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
5291 022346 104441 MOV #PRI05,R0
5292 022352 TRAP C$SPRI
5293 000004 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5294 022354 012767 000004 157672 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (42)
5295 022362 012767 177777 157660 MOV #1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
5296 022370 012767 000001 161452 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5297 022376 012767 010151 161446 MOV #4201,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
5298 022404 012767 005643 161442 MOV #EM4201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRtbl.
5299 022412 012767 011762 161436 MOV #ER0503,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
5300 ;+
5301 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
5302 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
5303 ; THIS SUBROUTINE REPORTS ERROR >>>> 4201 <<<<.
5304 022420 004767 171572 JSR PC,CLNRST ;RESET THE DHU-11. REPORT ANY ERRORS FOUND.
5305 022424 103120 BCC 60$ ;RESET FAILURE?, ABORT THIS TEST.
5306 ;+
5307 ; SET INTERNAL LOOPBACK,ENABLE RECEIVER FUNCTIONS ON ALL LINES.
5308 ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
5309 ; 2 STOP BITS.
5310 ; ENABLE TRANSMITTERS ON ALL LINES.
5311 ;-
5312 022426 004767 172036 JSR PC,FINACT ;FIND AN ACTIVE LINE.
5313 022432 103115 BCC 60$ ;EXIT THE TEST IF NO ACTIVE LINES.
5314 022434 010104 MOV R1,R4 ;SAVE THE LINE NUMBER.
5315 022436 012700 000204 MOV #204,R0 ;PASS THE LNCTRL CONTENTS.
5316 022442 004767 175104 JSR PC,WTWLNC ;INITIALISE THE LNCTRL REGISTERS.
5317 022446 012700 177670 MOV #177670,R0 ;PASS THE LPR CONTENTS.
5318 022452 004767 175124 JSR PC,WTWLPR ;INITIALSE THE LPR REGISTERS ON THE ACTIVE LINE.
5319 022456 004767 174440 JSR PC,TXENBL ;ENABLE TRANSMITTERS ON THE ACTIVE LINE.
5320 ;+
5321 ; VERIFY THAT THE DMA.START BIT IS CLEAR BEFORE ATTEMPTING THE DMA TRANSFER.
5322 ;-
5323 022462 005267 161364 INC ERRNBR ;SET THE ERROR NUMBER TO 4202.
5324 022466 032777 000200 157546 BIT #BIT07,BTXAD2A ;TEST THE DMA START BIT.
5325 022474 001072 BNE 50$ ;BRANCH TO REPORT THE ERROR IF THE BIT IS SET.
5326 ;+
5327 ; SET UP THE DMA REGISTERS TO PERFORM THE TRANSFER FROM THE ADDRESS OF THE CSR.
5328 ;-

```

```

5329 022476 016777 157524 157534      MOV    CSRA,0TXAD1A      ;SET UP THE LOW 16 BITS OF THE DMA ADDR.
5330 022504 012777 000001 157532      MOV    #1,0TXBFCA      ;SET UP TO DMA ONE CHARACTER.
5331 022512 112777 000203 157522      MOV    #203,0TXAD2A    ;SET UP THE 2 MSB'S AND INITIATE THE DMA.
5332 022520 012701 170012              MOV    #170012,R1        ;TEST BIT 15, TIME OUT OF 10 MS.
5333 022524 016702 157476              MOV    CSRA,R2          ;INDICATE TO TEST THE CSR.
5334 022530 005267 161316              INC     ERRNBR          ;SET THE ERROR NUMBER TO 4203.
5335 022534 004767 174676              JSR     PC,WAIBIS        ;WAIT FOR A TX-ACTION.
5336 022540 103050                      BCC     50$              ;REPORT THE ERROR IF NO TX-ACTION.
5337                                     ;+
5338                                     ; VERIFY THAT THE DMA ERROR BIT IS SET AND THE DMA START BIT IS CLEAR.
5339                                     ; -
5340 022542 005267 161304              INC     ERRNBR          ;SET THE ERROR NUMBER TO 4204.
5341 022546 032777 010000 157452      BIT     #BIT12,0CSRA    ;TEST THE DMA ERROR BIT.
5342 022554 001436                      BEQ     2$              ;REPORT THE ERROR IF BIT IS CLEAR.
5343 022556 005267 161270              INC     ERRNBR          ;SET THE ERROR NUMBER TO 4205.
5344 022562 032777 000200 157452      BIT     #BIT07,0TXAD2A  ;TEST THE DMA-START BIT.
5345 022570 001034                      BNE     50$              ;REPORT THE ERROR IF THE BIT IS SET.
5346                                     ;+
5347                                     ; VERIFY THAT THE DMA ERROR BIT CLEARS WHEN A "GOOD" DMA TRANSFER IS PERFORMED.
5348                                     ; -
5349 022572 010401                      MOV     R4,R1            ;SET UP THE ACTIVE LINE NUMBER.
5350 022574 012702 002650              MOV     #0UFBAS,R2      ;SET UP THE START ADDRESS OF THE DMA BUFFER.
5351 022600 012703 000001              MOV     #1,R3            ;SET UP TO DMA ONE CHARACTER.
5352 022604 005267 161242              INC     ERRNBR          ;SET THE ERROR NUMBER TO 4206.
5353 022610 004767 171564              JSR     PC,0DDMA        ;START THE DMA.
5354 022614 103022                      BCC     50$              ;REPORT THE ERROR IF ONE OCCURED.
5355 022616 012701 170036              MOV     #170036,R1      ;SET UP TO TEST BIT15 WITH TIMEOUT OF 30 MS.
5356 022622 016702 157400              MOV     CSRA,R2          ;INDICATE TO TEST THE CSR.
5357 022626 005267 161220              INC     ERRNBR          ;SET THE ERROR NUMBER TO 4207.
5358 022632 004767 174600              JSR     PC,WAIBIS        ;WAIT FOR A TX-ACTION.
5359 022636 103011                      BCC     50$              ;REPORT THE ERROR IF NO TX-ACTION OCCURED.
5360 022640 005267 161206              INC     ERRNBR          ;SET THE ERROR NUMBER TO 4208.
5361 022644 032702 010000              BIT     #BIT12,R2        ;TEST THE DMA ERROR BIT OF THE LAST CSR WORD READ.
5362 022650 001406                      BEQ     60$              ;EXIT THE TEST IF THE BIT IS CLEAR.
5363                                     ;+
5364                                     ; REPORT THE ERROR, DMA ERROR BIT BAD.
5365                                     ; -
5366 022652 012701 005675      2$:    MOV     #EM4202,R1      ;SET THE MESSAGE
5367                                     ; "DMA ERROR BIT BAD".
5368 022656                      ERROR                                     TRAP     C$ERROR
5369 022660 000402                      BR      60$              ;EXIT THE TEST.
5370
5371 022662 004767 174004      50$:    JSR     PC,TSABRT      ;REPORT THE NON-RELATED TEST ERROR.
5372 022666 005067 157356      60$:    CLR     CTRLCP        ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5373
5374 022672                      ENDTST
022672
022672 104401                      L10031: TRAP     C$ETST

```

B12

CZDHVBO DMU 11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 109
HARDWARE TEST - OAUTOI

SEQ 144

CZD
HAR

```

5376 .SBTTL HARDWARE TEST - OAUTOI -
5377 ;*****
5378 ;
5379 ;
5380 ; THIS TEST VERIFIES THAT THE DUT'S OAUTO FUNCTION BEHAVES CORRECTLY
5381 ; WHEN INACTIVE, IE OAUTO BIT CLEAR.
5382 ; THIS TEST WILL ONLY EXECUTE IF STAGGERED LOOPBACK MODE IS SELECTED.
5383 ; THE SPECIAL STAGGERED LOOPBACK CONNECTOR MUST BE FITTED.
5384 ;
5385 ;-----*****
5386
5387 BGNTEST
5388 022674 126727 157322 000002 CMPB LOPBCK,02 ;CHECK MODE SELECTED.
5389 022702 001402 BEQ .+6 ;DO NOT EXIT IF STAGGERED LOPBCK MODE SELECTED.
5390 022704 000167 000556 JMP 601 ;EXIT THIS TEST.
5391 022710 022710 000240 SETPRI @PRI05 ;ALLOW LTC INTERRUPTS.
5392 022714 104441 MOV @PRI05,RO
5393 022716 000005 TRAP C1SPRI
5394 022724 012767 177777 157330 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5395 022732 012767 000001 161110 MOV @TNUM,ISTNUM ;SET UP THE TEST NUMBER. (49)
5396 022740 012767 011445 161104 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
5397 022746 012767 005717 161100 MOV @1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5398 022754 012767 013136 161074 MOV @4901,ERRNBR ;SET ERROR NUMBER TO 4901.
5399 ;
5400 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
5401 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
5402 ; THIS SUBROUTINE REPORTS ERROR >>>> 4901 <<<<.
5403 ;
5404 022762 004767 171230 JSR PC,CLRST ;RESET THE DMU 11, REPORT ANY ERRORS FOUND.
5405 022766 103402 BCS .+6 ;DO NOT EXIT IF RESET WAS SUCCESSFUL.
5406 022770 000167 000472 JMP 601 ;EXIT THIS TEST.
5407 ;
5408 ; SET UP THE ASSOCIATED TX/RX LINE NUMBER TABLES.
5409 ;
5410 022774 004767 170442 JSR PC,ASLNTL ;INITIALISE THE ASSOCIATED TX/RX TABLES.
5411 ;
5412 ; SET EXTERNAL LOOPBACK, DISABLE OAUTO AND ENABLE RECEIVER ON ALL ACTIVE LINES.
5413 ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
5414 ; 2 STOP BITS.
5415 ; ENABLE TRANSMITTERS ON ALL LINES.
5416 ;
5417 023000 016705 157214 MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
5418 023004 012700 000004 MOV @4,RO ;PASS THE LNCRTL CONTENTS.
5419 023010 004767 174536 JSR PC,WTWLC ;INITIALISE THE LNCRTL REGISTERS.
5420 023014 012705 177777 MOV @MAPLNS,R5 ;PASS BIT MAP OF ALL LINES.
5421 023020 012700 177670 MOV @177670,RO ;PASS THE LPR CONTENTS.
5422 023024 004767 174552 JSR PC,WTWLP ;INITIALISE THE LPR REGISTERS ON ALL LINES.
5423 023030 004767 174066 JSR PC,TXENBL ;ENABLE TRANSMITTERS ON ALL LINES.
5424 ;
5425 ; SET UP OUTER LOOP FOR TESTING ACTIVE LINES IN BOTH LINE GROUPS.
5426 ;
5427 023034 012705 100000 MOV @100000,R5 ;SET UP LOOP CONTROL FLAG.
5428 023040 016705 157154 MOV ACTLNS,R5 ;GET THE ACTIVE LINE BIT MAP.
5429 023044 046705 157212 BIC LGPR2M,R5 ;REMOVE LINES IN GROUP 2.

```

C12

CZDHV80 DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 109-1
HARDWARE TEST - OAUTUI

SEQ 145

```

5430 023050 010567 000404      2:      MOV      R5,451      ;SAVE THE CURRENT LINE GROUP.
5431 023054 005067 000376      4:      CLR      401      ;CLEAR THE LINE NUMBER COUNTER.
5432 023060 016701 000372      4:      MOV      401,R1      ;COPY THE LINE NUMBER.
5433 023064 000241              CLC      ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5434 023066 006005              ROR      R5      ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5435 023070 103064              BCC      81      ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5436
5437      ;
5438      ; TEST THE STATE OF THE OAUTO BIT ON THE LINE UNDER TEST.
5439      ; REPORT ERROR IF IT IS FOUND SET, AND SKIP FURTHER TESTING OF THAT LINE.
5440 023072 012767 011446 160752      MOV      4902,ERRNBR      ;SET THE ERROR NUMBER TO 4902.
5441 023100 010177 157122              MOV      R1,0CSRA      ;SELECT THE LINE TO BE TESTED.
5442 023104 032777 000020 157124      BIT      0BIT4,0LNCTRA      ;TEST THE STATE OF THE OAUTO BIT.
5443 023112 001410              BEQ      61      ;SKIP ERROR REPORT IF OAUTO BIT IS CLEAR.
5444 023114 012702 005760              MOV      0EM4902,R2      ;PASS THE ERROR MESSAGE.
5445
5446 023120              ERROR      ; "OAUTO BIT BAD ON LINE NN"
5447      ;          >>>>> ERROR 04902 <<<<<,
5448      ;          TRAP      C1ERROR
5449
5450      ;
5451 023122 032767 000100 157060      ;
5452 023130 001556              ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5453
5454      ;
5455 023132 000443              BIT      0BIT06,OPTION      ;EXIT WITH TEST FAILURE MESSAGE IF
5456      ;          BEQ      601      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5457      ;          ;DURING THE SOFTWARE QUESTIONS.
5458
5459      ;
5460 023134 116177 004010 157064 6:      BR      81      ;SKIP FURTHER TESTING OF THIS LINE.
5461      ;
5462      ; TRANSMIT THE XOFF (ASCII DC3) ON THE ASSOCIATED LINE.
5463      ;
5464 023150 005267 160676      ;
5465 023154 012701 170012      ;
5466 023160 016702 157042      ;
5467 023164 004767 174246      ;
5468 023170 103134              ;
5469 023172 012704 000005      ;
5470 023176 004767 171060      ;
5471      ;
5472      ;
5473      ; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
5474      ; REPORT ERROR IF TX_ENABLE BIT IS CLEAR.
5475 023202 005267 160644      ;
5476 023206 016701 000244      ;
5477 023212 010177 157010      ;
5478 023216 005777 157020      ;
5479 023222 100407              ;
5480 023224 012702 005760      ;
5481      ;
5482 023230              ;
5483      ;          >>>>> ERROR 04904 <<<<<,
5484      ;          TRAP      C1ERROR

```

D12

```

5485                                     ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5486                                     ;
5487 023232 032767 000100 156750          BIT    #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
5488 023240 001512                      BEQ     60$          ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5489                                     ;DURING THE SOFTWARE QUESTIONS.
5490
5491 023242 005267 000210          8$:    INC     40$          ;INCREMENT THE LINE NUMBER.
5492 023246 005705                      TST     R5           ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5493 023250 001303                      BNE     4$           ;
5494
5495                                     ;+
5496                                     ; DISABLE TRANSMITTERS ON THE SELECTED LINES IN THE CURRENT LINE GROUP.
5497                                     ;-
5497 023252 016705 000202          MOV     45$,R5           ;RESTORE THE CURRENT LINE ACTIVE LINE GROUP.
5498 023256 004767 173544          JSR     PC,TXDSL          ;DISABLE TRANSMITTERS ON THE SELECTED LINES.
5499 023262 016705 000172          MOV     45$,R5           ;GET THE CURRENT ACTIVE LINE GROUP AGAIN.
5500 023266 005067 000164          CLR     40$           ;CLEAR THE LINE COUNTER.
5501 023272 012767 011451 160552 10$:    MOV     #4905,ERRNBR ;SET ERROR NUMBER TO 4905.
5502 023300 016701 000152          MOV     40$,R1          ;COPY THE LINE NUMBER.
5503 023304 000241                      CLC                ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5504 023306 006005                      ROR     R5           ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5505 023310 103041                      BCC     12$          ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5506
5507                                     ;+
5508                                     ; TRANSMIT THE XON (ASCII DC1) ON THE ASSOCIATED LINE.
5509                                     ;-
5509 023312 116177 004010 156706          MOVB   TXRLNB(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
5510 023320 112777 000021 156706          MOVB   #21,@DATA      ;TRANSMIT THE XON CHARACTER TO THE LUT.
5511
5512                                     ;+
5513                                     ; WAIT FOR TRANSMISSION TO COMPLETE.
5514                                     ;-
5514 023326 012701 170012          MOV     #170012,R1        ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
5515 023332 016702 156670          MOV     CSRA,R2          ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5516 023336 004767 174074          JSR     PC,WAIBIS         ;WAIT FOR TRANSMISSION TO COMPLETE.
5517 023342 103047                      BCC     50$          ;ABORT TEST IF TIMEOUT OCCURRED.
5518 023344 012704 000005          MOV     #5,R4           ;PASS TIME-OUT OF 5 MILLI SECS.
5519 023350 004767 170706          JSR     PC,DELAY         ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5520
5521                                     ;+
5522                                     ; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
5523                                     ; REPORT ERROR IF TX_ENABLE BIT IS SET.
5524                                     ;-
5524 023354 005267 160472          INC     ERRNBR           ;INCREMENT ERROR NUMBER TO 4906.
5525 023360 016701 000072          MOV     40$,R1          ;GET THE NUMBER OF THE LINE UNDER TEST.
5526 023364 010177 156636          MOV     R1,@CSRA        ;SELECT THE LINE CURRENTLY UNDER TEST.
5527 023370 005777 156646          TST     @TXAD2A         ;TEST THE STATE OF THE TX_ENABLE BIT.
5528 023374 100007                      BPL     12$          ;SKIP ERROR REPORT IF BIT IS CLEAR.
5529 023376 012702 005760          MOV     #EM4902,R2        ;PASS THE MESSAGE TO BE REPORTED.
5530                                     ; "OAUO BIT BAD ON LINE NN".
5531 023402                                     ;
5531 023402 104460          ERROR                                     ; >>>> ERROR #4906 <<<<<.
5532                                     ; TRAP C$ERROR
5533
5534                                     ;+
5535                                     ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5536                                     ;-
5536 023404 032767 000100 156576          BIT    #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
5537 023412 001425                      BEQ     60$          ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5538                                     ;DURING THE SOFTWARE QUESTIONS.
5539
5539 023414 005267 000036          12$:    INC     40$          ;INCREMENT THE LINE NUMBER.

```

E12

CZDHV80 DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 109-3
HARDWARE TEST - OAUTOI

SEQ 147

5541	023420	005705		TST	R5		;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5542	023422	001323		BNE	10\$		
5543							
5544							
5545							
5546							
5547							
5548	023424	005703		TST	R3		;HAVE BOTH LINE GROUPS BEEN TESTED?.
5549	023426	001417		BEQ	60\$		;YES; THEN EXIT THIS TEST.
5550	023430	005003		CLR	R3		;NO; CLEAR THE LOOP CONTROL FLAG.
5551	023432	012705	177777	MOV	MAPLNS,R5		;PASS THE BIT MAP OF ALL AVAILABLE LINE.
5552	023436	004767	173460	JSR	PC, TXENBL		;RE-ENABLE TRANSMISSION ON ALL LINES.
5553	023442	016705	156552	MOV	ACTLNS,R5		;GET THE ACTIVE LINE BIT MAP.
5554	023446	046705	156606	BIC	LGRP1M,R5		;REMOVE ALL ACTIVE LINES IN GROUP 1.
5555	023452	000167	177372	JMP	2\$		;ONCE MORE AROUND AND WE ARE DONE.
5556							
5557	023456	000000		40\$:	.WORD	0	;STORAGE FOR CURRENT LINE NUMBER.
5558	023460	000000		45\$:	.WORD	0	;STORAGE FOR CURRENT ACTIVE LINE BIT MAP.
5559	023462	004767	173204	50\$:	JSR	PC, TSABRT	;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5560	023466	005067	156556	60\$:	CLR	CTRLCF	;INDICATE THAT WE ARE NOT WITHIN A TEST.
5561							
5562	023472			ENDTST			
	023472						
	023472	104401					

L10032: TRAP C\$ETST

```

5564
5565
5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576 023474
5577 023474 126727 156522 000002
5578 023502 001402
5579 023504 000167 000556
5580 023510
5581 023510 012700 000240
5582 023514 104441
5583 023516 000006
5584 023516 012767 000006 156530
5585 023524 012767 177777 156516
5586 023532 012767 000001 160310
5587 023540 012767 011611 160304
5588 023546 012767 006012 160300
5589 023554 012767 013136 160274
5590
5591
5592
5593 023562 004767 170430
5594 023566 103402
5595 023570 000167 000472
5596
5597
5598
5599 023574 004767 167642
5600
5601
5602
5603
5604
5605
5606 023600 016705 156414
5607 023604 012700 000024
5608 023610 004767 173736
5609 023614 012705 177777
5610 023620 012700 177670
5611 023624 004767 173752
5612 023630 004767 173266
5613
5614
5615
5616 023634 012703 100000
5617 023640 016705 156354

```

```

.SBTTL  HARDWARE TEST          - OAUTOA -
;*****
;
;      - OAUTO BIT ACTIVE TEST -
;
;      THIS TEST VERIFIES THAT THE DUT'S OAUTO FUNCTION BEHAVES CORRECTLY
;      WHEN ACTIVE, IE OAUTO BIT ASSERTED HIGH.
;      THIS TEST WILL ONLY EXECUTE IF THE STAGGERED LOOPBACK MODE IS SELECTED.
;      THE SPECIAL STAGGERED LOOPBACK CONNECTOR MUST BE FITTED.
;*****
;-----
;
;      BGNTST
;
;      T6::
;      CMPB  LOPBCK, #2          ;CHECK MODE SELECTED.
;      BEQ   .+6                 ;DO NOT EXIT IF STAGGERD LOPBCK MODE SELECTED.
;      JMP   60$                 ;EXIT THIS TEST.
;      SETPRI #PRI05             ;ALLOW LTC INTERRUPTS.
;
;      MOV   #PRI05, R0
;      TRAP  C$SPRI
;
;      TNUM = TNUM + 1           ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
;      MOV   #TNUM, TSTNUM       ;SET UP THE TEST NUMBER. (50)
;      MOV   #-1, CTRLCF         ;INDICATE THAT WE ARE IN A TEST.
;      MOV   #1, ERRTP           ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
;      MOV   #5001, ERRNBR       ;SET ERROR NUMBER TO 5001.
;      MOV   #EM5001, ERRMSG      ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
;      MOV   #ER9101, ERRBLK     ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
;
;      ;
;      ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
;      ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
;      ; THIS SUBROUTINE REPORTS ERROR >>>> 5001 <<<<.
;      ;
;      JSR   PC, CLNRST          ;RESET THE DMU-11, REPORT ANY ERRORS FOUND.
;      BCS   .+6                 ;DO NOT EXIT IF RESET WAS SUCCESSFUL.
;      JMP   60$                 ;EXIT THIS TEST.
;
;      ;
;      ; SET-UP THE ASSOCIATED TX/RX LINE NUMBER TABLES.
;      ;
;      JSR   PC, ASLNTL          ;INITIALISE THE ASSOCIATED TX/RX TABLES.
;
;      ;
;      ; SET EXTERNAL LOOPBACK, ENABLE OAUTO AND RECEIVER FUNCTIONS ON ALL ACTIVE LINES
;      ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
;      ; 2 STOP BITS.
;      ; ENABLE TRANSMITTERS ON ALL LINES.
;      ;
;      MOV   ACTLNS, R5          ;PASS THE ACTIVE LINE BIT MAP.
;      MOV   #24, R0             ;PASS THE LNCTRL CONTENTS.
;      JSR   PC, WTWLNC          ;INITIALISE THE LNCTRL REGISTERS.
;      MOV   #MAPLNS, R5         ;PASS BIT MAP OF ALL LINES.
;      MOV   #177670, R0         ;PASS THE LPR CONTENTS.
;      JSR   PC, WTWLPR          ;INITIALISE THE LPR REGISTERS ON ALL LINES.
;      JSR   PC, TXENBL          ;ENABLE TRANSMITTERS ON ALL LINES.
;
;      ;
;      ; SET UP OUTER LOOP FOR TESTING ACTIVE LINES IN BOTH LINE GROUPS.
;      ;
;      MOV   #100000, R3         ;SET-UP LOOP CONTROL FLAG.
;      MOV   ACTLNS, R5          ;GET THE ACTIVE LINE BIT MAP.

```

```

5618 023644 046705 156412      BIC    LGRP2M,R5      ;REMOVE LINES IN GROUP 2.
5619 023650 010567 000404      2$:    MOV    R5,45$      ;SAVE THE CURRENT LINE GROUP.
5620 023654 005067 000376      CLR    40$      ;CLEAR THE LINE NUMBER COUNTER.
5621 023660 016701 000372      4$:    MOV    40$,R1      ;COPY THE LINE NUMBER.
5622 023664 000241      CLC      ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5623 023666 006005      ROR    R5      ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5624 023670 103064      BCC    8$      ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5625
5626      ;*
5627      ; TEST THE STATE OF THE OAUTO BIT ON THE LINE UNDER TEST.
5628      ; REPORT ERROR IF IT IS FOUND CLEAR, AND SKIP FURTHER TESTING OF THAT LINE.
5629 023672 012767 011612 160152      MOV    #5002,,ERRNBR      ;SET THE ERROR NUMBER TO 5002.
5630 023700 010177 156322      MOV    R1,@CSRA      ;SELECT THE LINE TO BE TESTED.
5631 023704 032777 000020 156324      BIT    @BIT4,@LNCTRA      ;TEST THE STATE OF THE OAUTO BIT.
5632 023712 001010      BNE    6$      ;SKIP ERROR REPORT IF OAUTO BIT IS SET.
5633 023714 012702 005760      MOV    @EM4902,R2      ;PASS THE ERROR MESSAGE.
5634      ; "OAUTO BIT BAD ON LINE NN"
5635 023720      ERROR      ; >>>>> ERROR #5002 <<<<<.
5636      TRAP    C$ERROR
5637
5638      ;*
5639      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5640 023722 032767 000100 156260      BIT    @BIT06,OPTION      ;EXIT WITH TEST FAILURE MESSAGE IF
5641 023730 001556      BEQ    60$      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5642      ; DURING THE SOFTWARE QUESTIONS.
5643
5644 023732 000443      BR    8$      ;SKIP FURTHER TESTING OF THIS LINE.
5645
5646      ;*
5647      ; TRANSMIT THE XOFF (ASCII DC3) ON THE ASSOCIATED LINE.
5648 023734 116177 004010 156264 6$:    MOVB    TXRLNB(R1),@CSRA      ;SELECT THE ASSOCIATED TX LINE.
5649 023742 112777 000023 156264      MOVB    #23,@DATA      ;TRANSMIT THE XOFF CHARACTER TO THE LUT.
5650
5651      ;*
5652      ; WAIT FOR TRANSMISSION TO COMPLETE.
5653 023750 005267 160076      INC    ERRNBR      ;INCREMENT ERROR NUMBER TO 5003.
5654 023754 012701 170012      MOV    #170012,R1      ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
5655 023760 016702 156242      MOV    CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5656 023764 004767 173446      JSR    PC,WAIBIS      ;WAIT FOR TRANSMISSION TO COMPLETE.
5657 023770 103134      BCC    50$      ;ABORT TEST IF TIMEOUT OCCURRED.
5658 023772 012704 000005      MOV    #5,R4      ;PASS TIME-OUT OF 5 MILLI SECS.
5659 023776 004767 170260      JSR    PC,DELAY      ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5660
5661      ;*
5662      ; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
5663      ; REPORT ERROR IF TX_ENABLE BIT IS SET.
5664 024002 005267 160044      INC    ERRNBR      ;INCREMENT ERROR NUMBER TO 5004.
5665 024006 016701 000244      MOV    40$,R1      ;GET THE NUMBER OF THE LINE TEST.
5666 024012 010177 156210      MOV    R1,@CSRA      ;SELECT THE LINE CURRENTLY UNDER TEST.
5667 024016 005777 156220      TST    @TXAD2A      ;TEST THE STATE OF THE TX_ENABLE BIT.
5668 024022 100007      BPL    8$      ;SKIP ERROR REPORT IF BIT IS CLEAR.
5669 024024 012702 005760      MOV    @EM4902,R2      ;PASS THE MESSAGE TO BE REPORTED.
5670      ; "OAUTO BIT BAD ON LINE NN".
5671 024030      ERROR      ; >>>>> ERROR #5004 <<<<<.
5672      TRAP    C$ERROR

```

```

5673
5674
5675
5676 024032 032767 000100 156150
5677 024040 001512
5678
5679
5680 024042 005267 000210 8$: INC 40$ ;INCREMENT THE LINE NUMBER.
5681 024046 005705 TST R5 ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5682 024050 001303 BNE 4$ ;
5683
5684
5685
5686 024052 016705 000202
5687 024056 004767 172744
5688 024062 016705 000172
5689 024066 005067 000164
5690 024072 012767 011615 157752 10$: MOV #5005,ERRNBR ;SET ERROR NUMBER TO 5005.
5691 024100 016701 000152 MOV 40$,R1 ;COPY THE LINE NUMBER.
5692 024104 000241 CLC ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5693 024106 006005 ROR R5 ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5694 024110 103041 BCC 12$ ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5695
5696
5697
5698 024112 116177 004010 156106
5699 024120 112777 000021 156106
5700
5701
5702
5703 024126 012701 170012
5704 024132 016702 156070
5705 024136 004767 173274
5706 024142 103047
5707 024144 012704 000005
5708 024150 004767 170106
5709
5710
5711
5712
5713 024154 005267 157672
5714 024160 016701 000072
5715 024164 010177 156036
5716 024170 005777 156046
5717 024174 100407
5718 024176 012702 005760
5719
5720 024202
024202 104460
5721
5722
5723
5724
5725 024204 032767 000100 155776
5726 024212 001425
5727
5728

;+
; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
;-
BIT #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
BEQ 60$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
;DURING THE SOFTWARE QUESTIONS.

;+
; DISABLE TRANSMITTERS ON THE SELECTED LINES IN THE CURRENT LINE GROUP.
;-
MOV 45$,R5 ;RESTORE THE CURRENT LINE ACTIVE LINE GROUP.
JSR PC,TXDSBL ;DISABLE TRANSMITTERS ON THE SELECTED LINES.
MOV 45$,R5 ;GET THE CURRENT LINE ACTIVE LINE GROUP AGAIN.
CLR 40$ ;CLEAR THE LINE COUNTER.
MOV #5005,ERRNBR ;SET ERROR NUMBER TO 5005.
MOV 40$,R1 ;COPY THE LINE NUMBER.
CLC ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
ROR R5 ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
BCC 12$ ;SKIP TESTING THIS LINE IF IT IS INACTIVE.

;+
; TRANSMIT THE XON (ASCII DC1) ON THE ASSOCIATED LINE.
;-
MOVB TXRLNB(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
MOVB #21,@FDATA ;TRANSMIT THE XON CHARACTER TO THE LUT.

;+
; WAIT FOR TRANSMISSION TO COMPLETE.
;-
MOV #170012,R1 ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
MOV CSRA,R2 ;PASS THE ADDRESS OF THE REGISTER TO TEST.
JSR PC,WAIBIS ;WAIT FOR DMA TO COMPLETE.
BCC 50$ ;ABORT TEST IF TIMEOUT OCCURRED.
MOV #5,R4 ;PASS TIME-OUT OF 5 MILLI SECS.
JSR PC,DELAY ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.

;+
; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
; REPORT ERROR IF TX_ENABLE BIT IS CLEAR.
;-
INC ERRNBR ;INCREMENT ERROR NUMBER TO 5006.
MOV 40$,R1 ;GET THE NUMBER OF THE LINE UNDER TEST.
MOV R1,@CSRA ;SELECT THE LINE CURRENTLY UNDER TEST.
TST @TXAD2A ;TEST THE STATE OF THE TX_ENABLE BIT.
BMI 12$ ;SKIP ERROR REPORT IF BIT IS SET.
MOV #EM4902,R2 ;PASS THE MESSAGE TO BE REPORTED.
; "OAUTO BIT BAD ON LINE NN".
; >>>>> ERROR #5006 <<<<<.
; TRAP C$ERROR

;+
; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
;-
BIT #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
BEQ 60$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
;DURING THE SOFTWARE QUESTIONS.

```

```

5729 024214 005267 000036      12$:      INC      40$      ;INCREMENT THE LINE NUMBER,
5730 024220 005705              TST      R5      ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5731 024222 001323              BNE      10$      ;
5732                          ;*
5733                          ; CHECK LOOP CONTROL FLAG TO DETERMINE IF BOTH SETS OF LINES HAVE BEEN TESTED
5734                          ; IF THIS IS THE FIRST TIME AROUND, RE-ENABLE TX ON ALL LINES, GENERATE ACTIVE
5735                          ; BIT MAP FOR SECOND LINE GROUP.
5736                          ;*
5737 024224 005703              TST      R3      ;HAVE BOTH LINE GROUPS BEEN TESTED?.
5738 024226 001417              BEQ      60$      ;YES; THEN EXIT THIS TEST.
5739 024230 005003              CLR      R3      ;NO; CLEAR THE LOOP CONTROL FLAG.
5740 024232 012705 177777      MOV      MAPLNS,R5 ;PASS THE BIT MAP OF ALL AVAILABLE LINE.
5741 024236 004767 172660      JSR      PC,TXENBL ;RE-ENABLE TRANSMISSION ON ALL LINES.
5742 024242 016705 155752      MOV      ACTLNS,R5 ;GET THE ACTIVE LINE BIT MAP.
5743 024246 046705 156006      BIC      LGRP1M,R5 ;REMOVE ALL ACTIVE LINES IN GROUP 1.
5744 024252 000167 177372      JMP      2$      ;ONCE MORE AROUND AND WE ARE DONE.
5745
5746 024256 000000      40$:      .WORD      0      ;STORAGE FOR CURRENT LINE NUMBER.
5747 024260 000000      45$:      .WORD      0      ;STORAGE FOR CURRENT ACTIVE LINE BIT MAP.
5748 024262 004767 172404      50$:      JSR      PC,TSABRT ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5749 024266 005067 155756      60$:      CLR      CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5750
5751 024272              ENDTST
      024272
      024272 104401

```

```

L10033:      TRAP      C$ETST

```

```

5753
5754 .SBTTL  HARDWARE TEST          - IAUTOI -
5755 ;*****
5756 ;*                               - IAUTO BIT INACTIVE TEST -
5757 ;*
5758 ;*   THIS TEST VERIFIES THAT THE DUT'S IAUTO FUNCTION BEHAVES CORRECTLY
5759 ;*   WHEN INACTIVE, IE. IAUTO BIT CLEAR.
5760 ;*   ALL ACTIVE LINES ARE TESTED INDIVIDUALLY BY FILLING THE FIFO
5761 ;*   THEN READING THE RECEIVED DATA CHECKING FOR THE PRESENCE OF
5762 ;*   XOFF(ASCII DC3) OR XON (ASCII DC1) CHARACTERS.
5763 ;*   IF ANY ARE FOUND THEN APPROPRIATE ERRORS ARE REPORTED.
5764 ;*   ANY BMP CODES THAT ARE FOUND WILL BE PLACED ON THE BMP CODE QUEUE,
5765 ;*   TO BE REPORTED LATER.
5766 ;*   THE CHARACTERS ARE TRANSMITTED ON ALL ACTIVE LINES, IN INTERNAL
5767 ;*   LOOPBACK MODE.
5768 ;*
5769 ;*****
5770
5771 024274      BGNTST
5772 024274      SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.      T7::
5773 024274 012700 000240      MOV      #PRI05,R0
5774 024300 104441      THAP      C$SPRI
5775 024302 012767 000007 155744      TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5776 024310 012767 177777 155732      MOV      #YNUM,TSTNUM      ;SET UP THE TEST NUMBER.      (51)
5777 024316 012767 000001 157524      MOV      #1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
5778 024324 012767 011755 157520      MOV      #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5779 024332 012767 006051 157514      MOV      #5101,ERRNBR      ;SET ERROR NUMBER TO 5101.
5780 024340 012767 013136 157510      MOV      #EM5101,ERRMSG      ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5781 ;*                               ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
5782 ;*
5783 ;*   RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
5784 ;*   CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
5785 ;*   THIS SUBROUTINE REPORTS ERROR >>>> 5101 <<<<.
5786 024346 004767 167644      JSR      PC,CLNRST      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
5787 024352 103156      BCC      60$      ;EXIT TEST IF FATAL ERROR FOUND.
5788
5789 ;*
5790 ;*   INITIALIZE THE 256 BYTE DATA PATTERN.
5791 ;*   ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
5792 ;*   NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
5793 024354 004767 170220      JSR      PC,INDTPX      ;INITIALISE DATA PATTERN.
5794
5795 ;*
5796 ;*   SET INTERNAL LOOPBACK, DISABLE IAUTO, ENABLE RECEIVER ON THE SELECTED LINE.
5797 ;*   SET LPR TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
5798 024360 016705 155634      MOV      ACTLNS,R5      ;PASS THE ACTIVE LINE BIT MAP.
5799 024364 012700 000204      MOV      #204,R0      ;PASS INT'L LOPBCK, ENABLE RX, DISABLE IAUTO.
5800 024370 004767 173156      JSR      PC,WTWLCN      ;INITIALISE THE LINE CONTROL REGISTER.
5801 024374 012700 177670      MOV      #177670,R0      ;PASS THE LPR CONTENTS.
5802 024400 004767 173176      JSR      PC,WIWLPR      ;SET THE LPR CONTENTS TO 38.4K BAUD.
5803 024404 012704 000012      MOV      #10,,R4      ;PASS DELAY TIME OF 10 MILLI SECONDS.
5804 024410 004767 167646      JSR      PC,DELAY      ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
5805
5806 ;*

```

K12

CZDHV80 DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 111-1
HARDWARE TEST - IAUTOI -

SEQ 153

```

5807      ; SET UP LOOP FOR ALL ACTIVE LINES.
5808      ; TEST THE STATE OF THE IAUTO BIT PRIOR TO TRANSMITTING THE DATA PATTERN.
5809      ; IF THE BIT IS SET, THEN REPORT THE ERROR AND SKIP TRANSMITTING
5810      ; THE DATA PATTERN ON THE SELECTED LINE.
5811      ; TRANSMIT A 256 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
5812      ; EMPTY THE FIFO, AND VERIFY NO XOFF OR XON CHARS WERE FOUND.
5813      ; -
5814 024414 005001      CLR      R1      ;CLEAR THE LINE NUMBER COUNTER.
5815 024416 005067 000264      CLR      55$      ;CLEAR STORAGE FOR LINE NUMBER.
5816 024422 012767 011756 157422 2$:      MOV      $5102,,ERRNBR      ;SET THE ERROR NUMBER TO 5102.
5817 024430 004767 170734      JSR      PC,PUFIFO      ;PURGE THE FIFO.
5818 024434 103121      BCC      50$      ;GO REPORT ERROR IF FIFO DID NOT PURGE.
5819 024436 000241      CLC      ;CLEAR CARRY PRIOR TO ROTATING BIT MAP.
5820 024440 006005      ROR      R5      ;ROTATE THE BIT MAP INTO THE CARRY BIT.
5821 024442 103107      BCC      12$      ;BRANCH IF LINE IS INACTIVE.
5822      ; -
5823      ; TEST THE IAUTO BIT ON THE SELECTED ACTIVE LINE.
5824      ; REPORT ERROR IF IT IS SET.
5825      ; DO NOT TRANSMIT THE DATA PATTERN ON THE SELECTED LINE.
5826      ; -
5827 024444 005267 157402      INC      ERRNBR      ;SET ERROR NUMBER TO 5103.
5828 024450 010177 155552      MOV      R1,$CSRA      ;SELECT LINE TO TEST.
5829 024454 032777 000002 155554      BIT      $BIT1,$LNCTRA      ;TEST THE STATE OF THE IAUTO BIT ON THIS LINE.
5830 024462 001410      BEQ      4$      ;SKIP ERROR IF IAUTO BIT CLEAR.
5831 024464 012702 006106      MOV      $EM5102,R2      ;PASS THE CORRECT ERROR MESSAGE.
5832 024470      ERROR      ;
5833      ; >>>>> ERROR <<<<<.
5834      ; TRAP      C$ERROR
5835      ; -
5836      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5837 024472 032767 000100 155510      BIT      $BIT06,OPTION      ;EXIT WITH TEST FAILURE MESSAGE IF
5838 024500 001503      BEQ      60$      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5839      ; DURING THE SOFTWARE QUESTIONS.
5840      ; -
5841 024502 000467      BR      12$      ;SKIP TRANSMITTING DATA PATTERN.
5842      ; -
5843      ; TRANSMIT DATA PATTERN OF 256 CHARS.
5844      ; -
5845      ; -
5846 024504 005267 157342 4$:      INC      ERRNBR      ;SET ERROR NUMBER TO 5104.
5847 024510 012702 002650      MOV      $BUF$AS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
5848 024514 012703 000400      MOV      $256,,R3      ;PASS THE LENGTH OF THE DATA PATTERN.
5849 024520 004767 167654      JSR      PC,DGDMA      ;TRANSMIT THE DATA PATTERN.
5850 024524 103065      BCC      50$      ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
5851      ; -
5852      ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER PLUS XOFF
5853      ; TO ARRIVE IN THE FIFO.
5854      ; -
5855      ; -
5856 024526 005267 157320      INC      ERRNBR      ;SET ERROR NUMBER TO 5105.
5857 024532 012701 170536      MOV      $170536,R1      ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
5858 024536 016702 155464      MOV      CSRA,R2      ;PASS THE ADDRESS OF THE CSR.
5859 024542 004767 172670      JSR      PC,WAIBIS      ;WAIT FOR DMA TO COMPLETE, TX ACTION SET.
5860 024546 103054      BCC      50$      ;IF NO TX ACTION WAS RECEIVED, ABORT THE TEST.
5861 024550 012704 000012      MOV      $10,,R4      ;PASS DELAY OF 10 MILLI SECS.
5862 024554 004767 167502      JSR      PC,DELAY      ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.

```

```

5863
5864
5865      ; READ 256 CHARS FROM THE FIFO.  REPORT ERROR IF ANY XOFF'S OR XON'S
5866      ; ARE FOUND.
5867      ;
5868      INC      ERRNBR      ;INCREMENT ERROR NUMBER TO 5106.
5869      MOV      0256,R1    ;INITIALISE THE READ COUNTER.
5870      MOV      0256,R2    ;READ CHAR FROM THE FIFO.
5871      BPL      50$        ;GO REPORT ERROR IF FIFO EMPTY.
5872      ;
5873      ; CHECK FOR BMP CODE IN THE FIFO.  SAVE ANY FOUND ON THE QUEUE.
5874      ;
5875      MOV      0170301,R0   ;SET UP BMP BIT MASK.
5876      BIC      R2,R0      ;TRY TO CLEAR ALL THE BMP BITS.
5877      BNE      8$         ;SKIP BMPSAV IF NOT A BMP CODE.
5878      JSR      PC,SAVBMP   ;SAVE THE BMP CODE ON THE QUEUE.
5879      ;
5880      ; CHECK FOR XOFF AND XON CHARACTERS.
5881      ;
5882      CMPB     R2,023      ;IS IT AN XOFF CHARACTER?.
5883      BEQ      10$        ;YES; GO REPORT ERROR.
5884      CMPB     R2,021      ;NO; IS IT AN XON CHARACTER?.
5885      BEQ      10$        ;YES; GO REPORT ERROR.
5886      DEC      R1         ;DECREMENT THE READ COUNT.
5887      BNE      6$         ;LOOP TO READ THE NEXT CHAR.
5888      BR       12$        ;GO CHECK FOR ANY UNTESTED ACTIVE LINES.
5889      ;
5890      INC      ERRNBR      ;SET ERROR NUMBER TO 5107.
5891      MOV      55$,R1      ;PASS THE LINE NUMBER TO BE REPORTED.
5892      MOV      05103,R2    ;PASS THE ERROR MESSAGE TO BE REPORTED.
5893      ERROR    >>>>> ERROR <<<<<.
5894      TRAP     C$ERROR
5895
5896      ;
5897      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
5898      BIT      06106,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
5899      BEQ      60$        ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
5900      ;DURING THE SOFTWARE QUESTIONS.
5901      ;
5902      ; CHECK IF ALL ACTIVE LINES HAVE BEEN TESTED.
5903      ;
5904      INC      55$         ;INCREMENT LINE NUMBER.
5905      MOV      55$,R1      ;GET NUMBER OF THE NEXT LINE TO TEST.
5906      TST      R5         ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
5907      BNE      2$         ;LOOP TO CHECK NEXT LINE.
5908      BR       60$        ;EXIT TEST.
5909      ;
5910      JSR      PC,TSABRT   ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5911      BR       60$        ;EXIT THIS TEST.
5912      ,WORD    0          ;STORAGE FOR LINE NUMBER.
5913      CLR      CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5914
5915      ENDTST

```

024650 104460
 024714 104401
 L10034: TRAP C\$ETST

```

5917 .SBTTL  HARDWARE TEST          - IAUTOA -
5918 ;*****
5919 ;*                                - IAUTO BIT ACTIVE TEST -
5920 ;*
5921 ;* THIS TEST VERIFIES THAT THE DUT'S IAUTO FUNCTION BEHAVES CORRECTLY
5922 ;* WHEN ACTIVE, IE IAUTO ASSERTED HIGH.
5923 ;* ALL ACTIVE LINES ARE TESTED INDIVIDUALLY BY FILLING THE FIFO, AND
5924 ;* CHECKING FOR THE PRESENCE OF AT LEAST ONE XOFF(ASCII DC3) CHARACTER
5925 ;* AND ONE XON (ASCII DC1) CHARACTER.
5926 ;* ANY BMP CODES THAT ARE FOUND WILL BE PLACED ON THE BMP CODE QUEUE,
5927 ;* TO BE REPORTED LATER.
5928 ;* THE CHARACTERS ARE TRANSMITTED ON ALL ACTIVE LINES, IN INTERNAL
5929 ;* LOOPBACK MODE.
5930 ;*
5931 ;*****
5932
5933 BGNTST
5934 024716 024716 T8::
5935 024716 012700 000240 SETPRI 0PRI05 ;ALLOW LTC INTERRUPTS.
5936 024722 104441 MOV 0PRI05,R0
5937 000010 TRAP C$SPRI
5938 024724 012767 000010 155322 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5939 024732 012767 177777 155310 MOV 0TNUM,TSTNUM ;SET UP THE TEST NUMBER. (52)
5940 024740 012767 000001 157102 MOV 0-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
5941 024746 012767 012121 157076 MOV 01,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5942 024754 012767 006174 157072 MOV 05201,,ERRNBR ;SET ERROR NUMBER TO 5201.
5943 024762 012767 013136 157066 MOV 0EM5201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5944 ;*
5945 ;* RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
5946 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
5947 ;* THIS SUBROUTINE REPORTS ERROR >>>> 5201 <<<<.
5948 ;*
5949 024770 004767 167222 JSR PC,CLNRST ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
5950 024774 103402 BCS 0+6
5951 024776 000167 000400 JMP 60$ ;EXIT TEST IF FATAL ERROR FOUND.
5952 ;*
5953 ;* INITIALIZE THE 256 BYTE DATA PATTERN.
5954 ;* ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
5955 ;* NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
5956 025002 004767 167572 JSR PC,INDTPX ;INITIALISE DATA PATTERN.
5957 ;*
5958 ;* SET INTERNAL LOOPBACK, ENABLE IAUTO AND RECEIVER ON THE SELECTED LINE.
5959 ;* SET LPR TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
5960 ;*
5961 025006 016705 155206 MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
5962 025012 012700 000206 MOV 0206,R0 ;PASS INTERNAL LOOPBACK, ENABLE RX AND IAUTO.
5963 025016 004767 172530 JSR PC,WTWLCN ;INITIALISE THE LINE CONTROL REGISTER.
5964 025022 012700 177670 MOV 0177670,R0 ;PASS THE LPR CONTENTS.
5965 025026 004767 172550 JSR PC,WTWLPR ;SET THE LPR CONTENTS TO 38.4K BAUD.
5966 025032 012704 000012 MOV 010,,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
5967 025036 004767 167220 JSR PC,DELAY ;WAIT FOR LNCRTL AND LPR REGS TO BE UPDATED.
5968 ;*
5969 ;*
5970 ;* SET UP LOOP FOR ALL ACTIVE LINES.

```

```

5971      ; TEST THE STATE OF THE OAUTO BIT PRIOR TO TRANSMITTING THE DATA PATTERN.
5972      ; IF THE BIT IS CLEAR, THEN REPORT THE ERROR AND SKIP TRANSMITTING
5973      ; THE DATA PATTERN ON THE SELECTED LINE.
5974      ; TRANSMIT A 224 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
5975      ; EMPTY THE FIFO, AND COUNT THE XOFF AND AN XON CHARS FOUND.
5976      ; -
5977 025042 005001      CLR      R1      ;CLEAR THE LINE NUMBER COUNTER.
5978 025044 005067 000330  CLR      55$      ;CLEAR STORAGE FOR LINE NUMBER.
5979 025050 012767 012122 156774 2$;  MOV      $5202,,ERRNBR  ;SET THE ERROR NUMBER TO 5202.
5980 025056 004767 170306      JSR      PC,PUFIFO  ;PURGE THE FIFO.
5981 025062 103143      BCC      50$      ;GO REPORT ERROR IF FIFO DID NOT PURGE.
5982 025064 000241      CLC      ;CLEAR CARRY PRIOR TO ROTATING BIT MAP.
5983 025066 006005      ROR      R5      ;ROTATE THE BIT MAP INTO THE CARRY BIT.
5984 025070 103131      BCC      16$      ;BRANCH IF LINE IS INACTIVE.
5985      ;+
5986      ; TEST THE IAUTO BIT ON THE SELECTED ACTIVE LINE.
5987      ; REPORT ERROR IF IT IS CLEAR.
5988      ; DO NOT TRANSMIT THE DATA PATTERN ON THE SELECTED LINE.
5989      ; -
5990 025072 005267 156754      INC      ERRNBR  ;SET ERROR NUMBER TO 5203.
5991 025076 010177 155124      MOV      R1,0CSRA  ;SELECT LINE TO TEST.
5992 025102 032777 000002 155126  BIT      $BIT1,$LNCTRA ;TEST THE STATE OF THE IAUTO BIT ON THIS LINE.
5993 025110 001010      BNE      4$      ;SKIP ERROR IF IAUTO BIT SET.
5994 025112 012702 006227      MOV      $EM5202,R2 ;PASS THE CORRECT ERROR MESSAGE.
5995      ; "IAUTO BIT FOUND CLEAR ON LINE NN"
5996 025116      ERROR      ; >>>> ERROR <<<<<
5997 025116 104460      TRAP      C$ERROR
5998
5999      ;+
6000      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
6001      ; -
6001 025120 032767 000100 155062  BIT      $BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
6002 025126 001525      BEQ      60$      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
6003      ; DURING THE SOFTWARE QUESTIONS.
6004
6005 025130 000511      BR      16$      ;SKIP TRANSMITTING DATA PATTERN.
6006
6007      ;+
6008      ; TRANSMIT DATA PATTERN TO FILL THE FIFO, 223 CHARS + 32 XOFF'S + XON.
6009      ; -
6010 025132 005267 156711 4$;  INC      ERRNBR  ;SET ERROR NUMBER TO 5204.
6011 025136 012702 002650      MOV      $BUFBAS,R2 ;PASS THE START OF THE DATA PATTERN TO TX.
6012 025142 012703 000337      MOV      $223,,R3  ;PASS THE LENGTH OF THE DATA PATTERN.
6013 025146 004767 167226      JSR      PC,DODMA  ;TRANSMIT THE DATA PATTERN.
6014 025152 103107      BCC      50$      ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6015
6016      ;+
6017      ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER PLUS XOFF
6018      ; TO ARRIVE IN THE FIFO.
6019      ; -
6020 025154 005267 156672      INC      ERRNBR  ;SET ERROR NUMBER TO 5205.
6021 025160 012701 170454      MOV      $170454,R1 ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6022 025164 016702 155036      MOV      CSRA,R2  ;PASS THE ADDRESS OF THE CSR.
6023 025170 004767 172242      JSR      PC,WAIBIS ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6024 025174 103076      BCC      50$      ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6025 025176 012704 000012      MOV      $10,,R4  ;PASS DELAY OF 10 MILLI SECS.
6026 025202 004767 167054      JSR      PC,DELAY  ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.

```

813

CZDHV80 DHU 11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 112-2
HARDWARE TEST - IAUTOA

SEQ 157

CZ
HA

```

6027
6028
6029
6030
6031 025206 005003
6032 025210 005004
6033 025212 005267 156634
6034 025216 012701 000400
6035 025222 017702 155002
6036 025226 100061
6037
6038
6039
6040 025230 012700 170301
6041 025234 040200
6042 025236 001002
6043 025240 004767 171234
6044
6045
6046
6047 025244 120227 000023
6048 025250 001001
6049 025252 005203
6050 025254 120227 000021
6051 025260 001001
6052 025262 005204
6053 025264 005301
6054 025266 001412
6055
6056
6057
6058
6059 025270 020127 000176
6060 025274 001352
6061 025276 010400
6062
6063 025300 012704 000001
6064 025304 004767 166752
6065 025310 010004
6066 025312 000743
6067
6068
6069
6070
6071 025314 005703
6072 025316 001403
6073 025320 020427 000001
6074 025324 001413
6075 025326 005267 156520
6076 025332 016701 000042
6077 025336 012702 006144
6078
6079 025342
      025342 104460
6080
6081
6082

      ; READ 256 CHARS FROM THE FIFO, COUNT ANY XOFF OR XON CHARS FOUND.
      ;
      CLR R3 ; CLEAR XOFF COUNTER.
      CLR R4 ; CLEAR XON COUNTER.
      INC ERRNR ; INCREMENT ERROR NUMBER TO 5206.
      MOV #256,R1 ; INITIALISE THE READ COUNTER.
61: MOV @RBUFA,R2 ; READ CHAR FROM THE FIFO.
      BPL 50$ ; GO REPORT ERROR IF FIFO EMPTY.
      ;
      ; CHECK FOR BMP CODE IN THE FIFO. SAVE ANY FOUND ON THE QUEUE.
      ;
      MOV #170301,R0 ; SET UP BMP BIT MASK.
      BIC R2,R0 ; TRY TO CLEAR ALL THE BMP BITS.
      BNE 81$ ; SKIP BMP SAV IF NOT A BMP CODE.
      JSR PC,SAVBMP ; SAVE THE BMP CODE ON THE QUEUE.
      ;
      ; CHECK FOR XOFF AND XON CHARACTERS.
      ;
81: CMPB R2,#23 ; IS IT AN XOFF CHARACTER?
      BNE 101$ ; NO. BRANCH TO SEE IF IT IS AN XON.
      INC R3 ; COUNT THE XOFF CHAR.
101: CMPB R2,#21 ; IS IT AN XON CHARACTER?
      BNE 121$ ; NO. SKIP THE NEXT INSTRUCTION.
      INC R4 ; COUNT THE XON.
121: DEC R1 ; DECREMENT THE READ COUNT.
      BEQ 131$ ; BRANCH IF ALL CHARACTERS READ.
      ;
      ; CHECK IF THE FIFO HAS BEEN EMPTIED BELOW THE HALF LEVEL. IF IT
      ; HAS DELAY FOR 1MS TO ALLOW THE XON TO BE GENERATED.
      ;
      CMP R1,#126. ; IS THE FIFO LEVEL = 126 ?
      BNE 61$ ; LOOP TO READ THE NEXT CHARACTER IF NOT.
      MOV R4,R0 ; SAVE THE XON COUNT, ALTHOUGH THERE SHOULDN'T
      ; BE ANY.
      MOV #1,R4 ; SET THE DELAY TO 1MS.
      JSR PC,DELAY ; PERFORM THE DELAY.
      MOV R0,R4 ; RESTORE THE XON COUNT.
      BR 61$ ; LOOP TO READ THE NEXT CHAR.
      ;
      ; VERIFY THAT AT LEAST 1 XOFF AND 1 XON WAS FOUND IN THE FIFO.
      ; REPORT ERROR IF NONE WERE FOUND.
      ;
131: LSL R3 ; CHECK XOFF COUNT.
      BEQ 141$ ; GO REPORT ERROR IF NONE FOUND.
      CMP R4,#1 ; CHECK XON COUNT = 1.
      BEQ 161$ ; SKIP THE ERROR REPORT IF ONE XON WAS FOUND.
141: INC ERRNR ; SET ERROR NUMBER TO 5207.
      MOV 55$,R1 ; PASS THE LINE NUMBER TO BE REPORTED.
      MOV @EM5103,R2 ; PASS THE ERROR MESSAGE TO BE REPORTED.
      ; "IAUTO BIT BAD ON LINE NN".
      ;
      ERROR ;
      ;
      ; ***** ERROR *****
      ;
      TRAP C1ERROR
      ;
      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

```

C13

CZDHVBO DHU 11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 112-3
 HARDWARE TEST IAUTOA

SEQ 158

```

6083
6084 025344 032767 000100 154636      BIT    0BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
6085 025352 001413                      BEQ    604 ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
6086                                     ;DURING THE SOFTWARE QUESTIONS.
6087
6088      ; CHECK IF ALL ACTIVE LINES HAVE BEEN TESTED.
6089
6090 025354 005267 000020      164:    INC    554 ;INCREMENT LINE NUMBER.
6091 025360 016701 000014      MOV    554,R1 ;GET NUMBER OF THE NEXT LINE TO TEST.
6092 025364 005705                      IST    R5 ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
6093 025366 001230                      BNE    24 ;LOOP TO CHECK NEXT LINE.
6094 025370 000404                      BR     604 ;EXIT TEST.
6095
6096 025372 004767 171274      504:    JSR    PC,ISABRT ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
6097 025376 000401                      BR     604 ;EXIT THIS TEST.
6098 025400 000000      554:    .WORD 0 ;STORAGE FOR LINE NUMBER.
6099 025402 005067 154642      604:    CLR    CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6100
6101 025406                      ENDTST
      025406
      025406 104401

```

L10035: TRAP C1ETST

```

6103 .SBTTL  HARDWARE TEST          - FIFDAT -
6104 .....
6105 ..... FIFO VALID DATA TEST .....
6106 .....
6107 .....
6108 ..... THIS TEST VERIFIES THAT THE DUT IS CAPABLE OF HOLDING 256 VALID
6109 ..... CHARACTERS IN ITS FIFO.
6110 ..... THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
6111 ..... INTERNAL LOOPBACK MODE.
6112 ..... THE DATA FOUND IN THE FIFO IS COMPARED WITH THE EXPECTED DATA, AND ANY
6113 ..... DISCREPANCIES ARE REPORTED.
6114 ..... ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
6115 ..... HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
6116 ..... REPORTED LATER.
6117 .....
6118 .....
6119 ..... BGNTEST
6120 .....          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.          T9::
6121 .....          MOV      #PRI05,R0
6122 .....          TRAP    C$SPRI
6123 .....          TNUM = TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6124 .....          MOV     #TNUM,TSTNUM     ;SET UP THE TEST NUMBER.          (53)
6125 .....          MOV     #-1,CTRLCF       ;INDICATE THAT WE ARE IN A TEST.
6126 .....          MOV     #1,ERRTYP        ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6127 .....          MOV     #5301,ERRNBR     ;SET ERROR NUMBER TO 5301.
6128 .....          MOV     #EM5301,ERRMSG    ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
6129 .....
6130 ..... ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
6131 ..... ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6132 ..... ; THIS SUBROUTINE REPORTS ERROR >>>> 5301 <<<<.
6133 .....
6134 .....          JSR      PC,CLNRST         ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6135 .....          BCC      60$              ;EXIT TEST IF FATAL ERROR FOUND.
6136 .....
6137 ..... ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
6138 ..... ; INITIALISE 256 BYTE DATA PATTERN.
6139 .....
6140 .....          JSR      PC,FINACT         ;FIND AN ACTIVE LINE.
6141 .....          BCC      60$              ;EXIT IF NO ACTIVE LINES FOUND.
6142 .....          JSR      PC,INDATP        ;INITIALISE THE DATA PATTERN.
6143 .....
6144 ..... ; TRANSMIT A 265 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
6145 ..... ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
6146 .....
6147 ..... ; SET INTERNAL LOOPBACK ON THE SELECTED LINE.
6148 ..... ; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
6149 .....
6150 .....          MOV     #204,R0           ;PASS PARAMETER FOR INTERNAL LOOPBACK,ENABLE RX.
6151 .....          JSR      PC,WTWLNCR       ;INITIALISE THE LINE CONTROL REGISTER.
6152 .....          MOV     #177670,R0        ;PASS THE LPR CONTENTS.
6153 .....          JSR      PC,WTWLPR        ;SET THE LPR CONTENTS TO 38.4K BAUD.
6154 .....          MOV     #10,R4           ;PASS DELAY TIME OF 10 MILLI SECONDS.
6155 .....          JSR      PC,DELAY         ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6156 .....          MOV     #BUFBAS,R2        ;PASS THE START OF THE DATA PATTERN TO IX.
6157 .....          MOV     #BUFMID-BUFBAS,R3 ;PASS THE LENGTH OF THE DATA PATTERN.

```

```

6157 025534 005267 156312      INC      ERRNBR      ;SET ERROR NUMBER TO 5302.
6158 025540 004767 166634      JSR      PC,DODMA    ;TRANSMIT THE DATA PATTERN.
6159 025544 103057              BCC      50$          ;ABORT TEST IF ERROR FOUND DURING DMA TX.
6160                               ;*
6161                               ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
6162                               ; THE FIFO.
6163                               ;*
6164 025546 005267 156300      INC      ERRNBR      ;SET ERROR NUMBER TO 5303.
6165 025552 010103              MOV      R1,R3        ;SAVE THE NUMBER OF THE SELECTED ACTIVE LINE.
6166 025554 012701 170536      MOV      #170536,R1    ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
6167 025560 016702 154442      MOV      CSRA,R2      ;PASS THE ADDRESS OF THE CSR.
6168 025564 004767 171646      JSR      PC,WAIBIS    ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6169 025570 103045              BCC      50$          ;BRANCH IF FIFO EMPTY, ABORT THE TEST.
6170 025572 012704 000005      MOV      #5,R4        ;PASS DELAY OF 5 MILLI SECS.
6171 025576 004767 166460      JSR      PC,DELAY     ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6172                               ;*
6173                               ; READ THE FIFO CHECKING FOR DATA CORRUPTION, REPORT ANY ERRORS FOUND.
6174                               ; ABORT THE TEST IF A BMP CODE WAS FOUND IN THE FIFO.
6175                               ;*
6176 025602 006303              ASL      R3            ;MULTIPLY BY 2.
6177 025604 005004              CLR      R4            ;INITIALISE THE EXPECTED DATA.
6178 025606 016705 154416      MOV      RBUFA,R5      ;GET THE ADDRESS OF THE RECEIVER BUFFER REG.
6179 025612 012767 012270 156232 2$: MOV      #5304,ERRNBR ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
6180 025620 011502              MOV      (R5),R2      ;GET THE ACTUAL DATA FROM THE FIFO.
6181 025622 100030              BPL      50$          ;ABORT THE TEST IF THE FIFO IS EMPTY.
6182                               ;*
6183                               ; CHECK IF THE READ CHARACTER IS A BMP CODE.
6184                               ; IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
6185                               ; ABORT THE TEST.
6186                               ;*
6187 025624 005267 156222      INC      ERRNBR      ;SET ERROR NUMBER TO 5305.
6188 025630 004767 166142      JSR      PC,CHKBMP    ;CHECK IF CHARACTER IS A BMP CODE.
6189 025634 103002              BCC      4$          ;BRANCH IF NOT A BMP CODE.
6190 025636              ERROR              ;>>>> ERROR 5305 <<<<<.
6191              104460                                TRAP      C$ERROR
6192 025640 000423              BR      60$          ;ABORT THIS TEST.
6193
6194 025642 005267 156204      4$: INC      ERRNBR      ;SET ERROR NUMBER TO 5306.
6195 025646 120402              CMPB     R4,R2        ;COMPARE THE EXPECTED WITH THE ACTUAL DATA.
6196 025650 001412              BEQ      8$          ;SKIP ERROR REPORT IF DATA IS OK.
6197 025652 012767 012644 156176      MOV      #ER9002,ERRBK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6198 025660 012701 006321      MOV      #EM5302,R1    ;PASS THE MESSAGE TO BE REPORTED.
6199              ;REPORT THE ERROR "FIFO BAD, DATA FIELD CORRUPTED"
6200 025664      6$: ERROR              ;>>>> ERROR 5306 <<<<<.
6201              104460                                TRAP      C$ERROR
6202
6203                               ;*
6204                               ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
6205                               ;*
6205 025666 032767 000100 154314      BIT      #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
6206 025674 001405              REQ      60$          ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
6207                               ; DURING THE SOFTWARE QUESTIONS.
6208
6209 025676 105204      8$: INCB      R4            ;INCREMENT THE EXPECTED DATA.
6210 025700 001344              BNE      2$          ;LOOP IF NOT DONE.
6211 025702 000402              BR      60$          ;EXIT

```

F13

CZDHV80 DHU 11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 113-2
HARDWARE TEST - FIFDAT

SEQ 161

6212
6213 025704 004767 170762 50\$: JSR PC,TSABRT ;ABORT THE TEST, REASON SHOWN BY ERROR NUMBER.
6214 025710 005067 154334 60\$: CLR CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6215
6216 025714
025714
025714 104401
ENDTST
L10036: TRAP C\$ETST

```

6218 .SBTTL HARDWARE TEST - FI3QLI -
6219 ;*****
6220 ; - FIFO 3/4 LEVEL INACTIVE TEST -
6221 ;
6222 ; THIS TEST VERIFIES THAT THE DUT'S FIFO 3/4 LEVEL ALARM SYSTEM
6223 ; REMAINS INACTIVE WHILE IT CONTAINS 191 CHARACTERS OR LESS.
6224 ; THE TEST LOOKS FOR AN XOFF (ASCII DC3) CHARACTER IN THE FIFO.
6225 ; IF ANY XOFF'S ARE FOUND AN ERROR WILL BE REPORTED AND THE TEST ABORTED.
6226 ; ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
6227 ; HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
6228 ; REPORTED LATER.
6229 ; THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
6230 ; INTERNAL LOOPBACK MODE.
6231 ;
6232 ; -*****
6233
6234 025716 BGNST
6235 025716 T10::
025716 SETPRI #PRIOS ;ALLOW LTC INTERRUPTS.
025722 012700 000240 MU' #PRIOS,RO
104441 TRAP C$PRI
000012 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
025724 012767 000012 154322 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (54)
025732 012767 177777 154310 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
025740 012767 000001 156102 MOV #1,ERRTYP ;SET FATAL ERROR TYPE IN ERROR TABLE.
025746 012767 012431 156076 MOV #5401,ERRNBR ;SET ERROR NUMBER TO 5401.
025754 012767 006452 156072 MOV #EM5401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
025762 012767 011762 156066 MOV #ER0503,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6243 ;
6244 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
6245 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6246 ; THIS SUBROUTINE REPORTS ERROR >>>> 5401 <<<<.
6247 ;
6248 025770 004767 166222 JSR PC,CLNRST ;RESET THE DMU-11, REPORT ANY ERRORS FOUND.
6249 025774 103111 BCC 60$ ;EXIT TEST IF FATAL ERROR FOUND.
6250 ;
6251 ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
6252 ;
6253 025776 004767 166466 JSR PC,FINACT ;FIND THE NUMBER OF THE FIRST ACTIVE LINE.
6254 026002 103106 BCC 60$ ;EXIT IF NO LINES ARE AVAILABLE.
6255 ;
6256 ;
6257 ; INITIALIZE THE 256 BYTE DATA PATTERN.
6258 ; ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
6259 ; NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
6260 ;
6261 026004 004767 166570 JSR PC,INDTPX ;INITIALISE THE DATA PATTERN.
6262 ;
6263 ; TRANSMIT A 191 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL.
6264 ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
6265 ;
6266 ;
6267 ; SET INTERNAL LOOPBACK, ENABLE IAUTO AND RX ON THE SELECTED LINE.
6268 ; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
6269 ;
6270 026010 012700 000206 MOV #206,RO ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
6271 026014 004767 171532 JSR PC,WTWLCN ;INITIALISE THE LINE CONTROL REGISTER.

```

```

6272 026020 012700 177670      MOV    #177670,R0      ;PASS THE LPR CONTENTS.
6273 026024 004767 171552      JSR    PC,WTWLPR    ;SET THE LPR CONTENTS TO 38.4K BAUD.
6274 026030 012704 000012      MOV    #10.,R4      ;PASS DELAY TIME OF 10 MILLI SECONDS.
6275 026034 004767 166222      JSR    PC,DELAY      ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6276 026040 012702 002650      MOV    #BUFBAS,R2    ;PASS THE START OF THE DATA PATTERN TO TX.
6277 026044 012703 000277      MOV    #191.,R3     ;PASS THE LENGTH OF THE DATA PATTERN.
6278 026050 004767 166324      JSR    PC,DODMA      ;TRANSMIT THE DATA PATTERN.
6279 026054 103057             BCC    50$           ;IF ERROR FOUND DURING DMA THEN ABORT TEST.
6280
6281
6282
6283
6284
6285 026056 005267 155770      INC    ERRNBR        ;SET ERROR NUMBER TO 5402.
6286 026062 012701 170454      MOV    #170454,R1    ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6287 026066 016702 154134      MOV    CSRA,R2      ;PASS THE ADDRESS OF THE CSR.
6288 026072 004767 171340      JSR    PC,WAIBIS     ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6289 026076 103046             BCC    50$           ;IF FIFO EMPTY, REPORT ERROR, ABORT THE TEST.
6290 026100 012704 000005      MOV    #5,R4        ;PASS DELAY OF 5 MILLI SECS.
6291 026104 004767 166152      JSR    PC,DELAY      ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6292
6293
6294
6295
6296
6297
6298
6299
6300 026110 005004             CLR    R4            ;CLEAR THE CHARACTER COUNT.
6301 026112 016705 154112      MOV    RBUFA,R5      ;GET THE ADDRESS OF THE RECEIVER BUFFER REG.
6302 026116 012767 012433 155726 2$: MOV    #5403.,ERRNBR    ;SET ERROR NUMBER TO 5403.
6303 026124 011502             MOV    (R5),R2          ;GET THE ACTUAL DATA FROM THE FIFO.
6304 026126 100032             BPL    50$           ;FIFO EMPTY, ABORT TEST.
6305 026130 005204             INC    R4            ;COUNT THE CHARACTER.
6306
6307
6308
6309
6310
6311 026132 005267 155714      INC    ERRNBR        ;SET ERROR NUMBER TO 5404.
6312 026136 004767 165634      JSR    PC,CHKBMP     ;CHECK IF CHARACTER IS A BMP CODE.
6313 026142 103001             BCC    4$           ;BRANCH IF NOT A BMP CODE.
6314
6315 026144 000421             BR      8$           ;REPORT THE ERROR "BMP CODE FOUND IN FIFO, TEST INVALIDATED".
6316
6317
6318
6319
6320 026146 005267 155700      INC    ERRNBR        ;SET ERROR NUMBER TO 5405.
6321 026152 122702 000023      CMPB   #23,R2       ;CHECK IF THE READ DATA IS AN XOFF.
6322 026156 001003             BNE    6$           ;BRANCH IF NOT AN XOFF.
6323 026160 012701 006520      MOV    #EM5402,R1    ;PASS THE MESSAGE TO BE REPORTED.
6324
6325 026164 000411             BR      8$           ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
6326
6327 026166 005267 155660      INC    ERRNBR        ;GO REPORT THE ERROR AND ABORT THE TEST.
6328 026172 020427 000277      CMP     R4,#191.     ;CHECK IF WE HAVE READ ALL THE CHARACTERS.

```

```

6329 026176 001347      BNE      2$          ;LOOP BACK TO GET THE NEXT CHARACTER.
6330 026200 011502      MOV      (R5),R2      ;TRY TO READ AN EXTRA CHARACTER FROM THE FIFO.
6331 026202 100006      BPL      60$          ;EXIT IF NON FOUND.
6332 026204 012701 006520  MOV      0EM5402,R1    ;PASS THE MESSAGE TO BE REPORTED.
6333                                     ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
6334
6335 026210                8$:      ERROR          ;          >>>> ERRORS 5304 THRU 5306 <<<<.
                                TRAP      C$ERROR
6336 026212 104460      BR      60$          ;EXIT THE TEST.
6337
6338                                     ;          >>>> ERRORS 5402 AND 5403 <<<<.
6339 026214 004767 170452 50$:      JSR      PC,ISABRT    ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
6340 026220 005067 154024 60$:      CLR      CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6341
6342 026224                ENDTST
                                L10037:
                                TRAP      C$ETST
                                026224 104401

```

```

6344 .SBTTL  HARDWARE TEST          - FI3QLA -
6345 ;*****
6346 ;
6347 ;
6348 ;
6349 ;
6350 ;
6351 ;
6352 ;
6353 ;
6354 ;
6355 ;
6356 ;
6357 ;
6358 ;
6359 ;
6360 ;
6361 026226      BGNTST
6362 026226      SETPRI  @PRI05          ;ALLOW LTC INTERRUPTS.
6363 026226      MOV      @PRI05,R0      T11::
6364 026232      TRAP     C@SPRI
6365 012700      TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6366 000013      MOV      @TNUM,TSTNUM   ;SET UP THE TEST NUMBER. (55)
6367 012767      MOV      @-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
6368 012767      MOV      @1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6369 012575      MOV      @5501,,ERRNBR   ;SET ERROR NUMBER TO 5501.
6370 006561      MOV      @EM5501,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
6371 ;
6372 ;
6373 ;
6374 026272      JSR      PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
6375 103402      BCS      .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
6376 000167      JMP      60$            ;EXIT TEST FATAL ERROR FOUND.
6377 ;
6378 ;
6379 ;
6380 026304      JSR      PC,FINACT        ;FIND AN ACTIVE LINE.
6381 103402      BCS      .+6             ;SKIP EXIT OF TEST IF ACTIVE LINE FOUND.
6382 000167      JMP      60$            ;EXIT TEST.
6383 ;
6384 ;
6385 ;
6386 ;
6387 ;
6388 026316      JSR      PC,INDTPX        ;INITIALISE DATA PATTERN.
6389 ;
6390 ;
6391 ;
6392 ;
6393 ;
6394 ;
6395 ;
6396 ;
6397 026322      INC      ERRNBR          ;SET ERROR NUMBER TO 5502.

```

6398	026326	012700	000206	MOV	#206,R0	;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
6399	026332	004767	171214	JSR	PC,WTWLCN	;INITIALISE THE LINE CONTROL REGISTER.
6400	026336	012700	177670	MOV	#177670,R0	;PASS THE LPR CONTENTS.
6401	026342	004767	171234	JSR	PC,WTWLCR	;SET THE LPR CONTENTS TO 38.4K BAUD.
6402	026346	012704	000012	MOV	#10.,R4	;PASS DELAY TIME OF 10 MILLI SECONDS.
6403	026352	004767	165704	JSR	PC,DELAY	;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6404	026356	010105		MOV	R1,R5	;COPY THE LINE NUMBER.
6405	026360	012702	002650	MOV	#BUFBA5,R2	;PASS THE START OF THE DATA PATTERN TO TX.
6406	026364	012703	000277	MOV	#191.,R3	;PASS THE LENGTH OF THE DATA PATTERN.
6407	026370	004767	166004	JSR	PC,DODMA	;TRANSMIT THE DATA PATTERN.
6408	026374	103147		BCC	50\$	;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6409						
6410						
6411						
6412						
6413						
6414	026376	005267	155450	INC	ERRNBR	;SET ERROR NUMBER TO 5503.
6415	026402	012701	170454	MOV	#170454,R1	;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6416	026406	016702	153614	MOV	CSRA,R2	;PASS THE ADDRESS OF THE CSR.
6417	026412	004767	171020	JSR	PC,WAIBIS	;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6418	026413	103136		BCC	50\$	;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6419	026419	012704	000005	MOV	#5,R4	;PASS DELAY OF 5 MILLI SECS.
6420	026424	004767	165632	JSR	PC,DELAY	;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6421						
6422						
6423						
6424	026430	005267	155416	INC	ERRNBR	;SET ERROR NUMBER TO 5504.
6425	026434	010501		MOV	R5,R1	;PASS THE LINE NUMBER.
6426	026436	012702	002650	MOV	#BUFBA5,R2	;PASS THE START OF THE DATA PATTERN TO TX.
6427	026442	012703	000001	MOV	#1,R3	;PASS THE NUMBER OF CHARACTERS TO TX.
6428	026446	004767	165726	JSR	PC,DODMA	;TX A NULL CHARACTER TO CAUSE AN XOFF.
6429	026452	103120		BCC	50\$	;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6430						
6431						
6432						
6433	026454	005267	155372	INC	ERRNBR	;SET ERROR NUMBER TO 5505.
6434	026460	012701	170012	MOV	#170012,R1	;PASS TIME-OUT VALUE OF 10 MILLI SECS.
6435	026464	016702	153536	MOV	CSRA,R2	;PASS THE ADDRESS OF THE CSR.
6436	026470	004767	170742	JSR	PC,WAIBIS	;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6437	026474	103107		BCC	50\$	;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6438	026476	012704	000005	MOV	#5,R4	;PASS DELAY OF 5 MILLI SECS.
6439	026502	004767	165554	JSR	PC,DELAY	;WAIT FOR XOFF TO GET INTO THE FIFO.
6440						
6441						
6442						
6443	026506	012702	002650	MOV	#BUFBA5,R2	;INITIALIZE THE DATA PATTERN TO BE
6444	026512	105022		CLRB	(R2),	; ALL NULLS.
6445	026514	020227	003250	CMP	R2,#BUFMID	
6446	026520	103774		BLO	4\$	
6447						
6448						
6449						
6450						
6451						
6452	026522	005267	155324	INC	ERRNBR	;SET ERROR NUMBER TO 5506.
6453	026526	010501		MOV	R5,R1	;PASS THE LINE NUMBER.
6454	026530	012702	002650	MOV	#BUFBA5,R2	;PASS THE START OF THE DATA PATTERN TO TX.

6455	026534	012703	000037			MOV	R3		;PASS THE LENGTH OF THE DATA PATTERN.
6456	026540	004767	165634			JSR	PC,DODMA		;TRANSMIT THE DATA PATTERN.
6457	026544	103063				BCC	50\$		;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6458									
6459									; WAIT FOR THE XOFF'S AND THE NULL CHARACTERS TO BE RECEIVED.
6460									; THERE ARE NOW 255 CHARACTERS IN THE FIFO.
6461									
6462	026546	005267	155300			INC	ERRNBR		;SET ERROR NUMBER TO 5507.
6463	026552	012701	170454			MOV	#170454,R1		;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6464	026556	016702	153444			MOV	CSRA,R2		;PASS THE ADDRESS OF THE CSR.
6465	026562	004767	170650			JSR	PC,WAIBIS		;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6466	026566	103052				BCC	50\$		;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6467	026570	012704	000005			MOV	#5,R4		;PASS DELAY OF 5 MILLI SECS.
6468	026574	004767	165462			JSR	PC,DELAY		;WAIT FOR XOFF TO GET INTO THE FIFO.
6469									
6470									; READ THE FIFO UNTIL EMPTY, COUNTING THE NUMBER OF XOFF CHARACTERS
6471									; THAT ARE FOUND.
6472									
6473	026600	005004				CLR	R4		;CLEAR CHARACTER COUNTER.
6474	026602	005003				CLR	R3		;CLEAR THE XOFF FOUND COUNTER.
6475	026604	012701	170001			MOV	#170001,R1		;INDICATE TO TEST DATA.VALID BIT, TIME-OUT 1MS.
6476	026610	012767	012604	155234	6\$:	MOV	#5508,,ERRNBR		;SET UP ERROR NUMBER EACH TIME AROUND THE LOOP.
6477	026616	016702	153406			MOV	RBUFA,R2		;INDICATE TO CHECK RECEIVE BUFFER REGISTER.
6478	026622	004767	17C610			JSR	PC,WAIBIS		;WAIT FOR RECEIVED CHAR OR TIME-OUT.
6479	026626	103032				BCC	50\$		;GO REPORT ERROR IF FIFO EMPTY.
6480	026630	005204				INC	R4		;COUNT THE CHARACTER.
6481									
6482									; CHECK FOR BMP CODES IN THE FIFO, ABORT THE TEST IF ANY ARE FOUND.
6483									; SAVE THE BMP CODE ON THE QUEUE TO BE REPORTED LATER.
6484									
6485	026632	005267	155214			INC	ERRNBR		;SET ERROR NUMBER TO 5509.
6486	026636	004767	165134			JSR	PC,CBKBP		;CHECK IF WE HAVE GOT A BMP CODE.
6487	026642	103422				BCS	12\$		;GO REPORT THE ERROR IF WE FOUND A BMP CODE.
6488									
6489									; CHECK FOR XOFF CHARACTER.
6490									
6491	026644	122702	000023		8\$:	CMPB	#23,R2		;CHECK IF THE RECEIVED CHARACTER WAS AN XOFF.
6492	026650	001001				BNE	10\$		;BRANCH IF CHARACTER WAS NOT AN XOFF.
6493	026652	005203				INC	R3		;INCREMENT XOFF FOUND COUNT.
6494									
6495									; CHECK IF ALL THE CHARACTERS INCLUDING THE XON HAVE BEEN REMOVED.
6496									
6497	026654	020427	000400		10\$:	CMP	R4,#256.		;CHECK IF WE HAVE REMOVED ALL THE CHARACTERS.
6498	026660	002753				BLT	6\$		;GO GET THE NEXT CHAR IF WE HAVE NOT FINISHED.
6499									
6500									
6501									; CHECK IF THE CORRECT NUMBER OF XOFF'S WERE FOUND IN THE FIFO,
6502									; REPORT ERROR IF COUNT IS INCORRECT.
6503									
6504	026662	012767	012606	155162		MOV	#5510,,ERRNBR		;SET UP THE ERROR NUMBER TO 5510.
6505	026670	022703	000040			CMP	#32,,R3		;COMPARE EXPECTED XOFF COUNT WITH ACTUAL COUNT.
6506	026674	001411				BEQ	60\$		;EXIT TEST IF SUCCESS.
6507	026676	012767	011762	155152		MOV	#E0503,ERRBLK		;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6508	026704	012701	006520			MOV	#EM5402,R1		;PASS THE MESSAGE TO BE REPORTED.
6509									REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
6510	026710				12\$:	ERROR		</	

M13

CZDHVBO DHU-11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 115-3
HARDWARE TEST - FI3QLA -

SEQ 168

```
6511 026712 000402          BR      60$      ;ABORT THE TEST.
6512
6513 026714 004767 167752    50$:    JSR      PC,TSABRT    ;REPORT TEST ABORTED. ERROR # SHOWS REASON.
6514 026720 005067 153324    60$:    CLR      CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6515
6516 026724          ENDTST
      026724
      026724 104401          L10040:
                                TRAP      C$ETST
```

```

6518 .SBTTL  HARDWARE TEST          - FI3QAI -
6519 :*****
6520 :
6521 :
6522 :
6523 :
6524 :
6525 :
6526 :
6527 :
6528 :
6529 :
6530 :
6531 :
6532 :
6533 :
6534 :
6535 :
6536 :
6537 :
6538 :
6539 :
6540 :
6541 :
6542 :
6543 :
6544 :
6545 :
6546 :
6547 :
6548 :
6549 :
6550 :
6551 :
6552 :
6553 :
6554 :
6555 :
6556 :
6557 :
6558 :
6559 :
6560 :
6561 :
6562 :
6563 :
6564 :
6565 :
6566 :
6567 :
6568 :
6569 :
6570 :
6571 :

```

```

        BGNTST
        SETPRI 0PRI05          ;ALLOW LTC INTERRUPTS.
                                T12::
                                MOV 0PRI05,R0
                                TRAP C$SPRI
        TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
        MOV 0TNUM,ISTNUM     ;SET UP THE TEST NUMBER. (56)
        MOV 0-1,CTRLCF       ;INDICATE THAT WE ARE IN A TEST.
        MOV 01,ERRTYP         ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
        MOV 05601,ERRNBR      ;SET ERROR NUMBER TO 5601.
        MOV 0EM5601,ERRMSG    ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
;
; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; THIS SUBROUTINE REPORTS ERROR >>>> 5601 <<<<.
;
        JSR PC,CLNRST         ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
        BCS 2$                ;SKIP EXITING TEST A SUCCESSFUL RESET.
        JMP 60$                ;EXIT THIS TEST.
2$:
;
; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
;
        JSR PC,FINACT          ;FIND AN ACTIVE LINE.
        BCS 16$                ;SKIP EXIT OF TEST IF ACTIVE LINE FOUND.
        JMP 60$                ;EXIT TEST.
;
; INITIALIZE THE 256 BYTE DATA PATTERN.
; ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
; NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
;
        JSR PC,INDTPX          ;INITIALISE THE DATA PATTERN.
;
; TRANSMIT A 256 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
;
;
; SET INTERNAL LOOPBACK, ENABLE IAUTO AND RECEIVER ON THE SELECTED LINE.
; TRANSMIT THE FIRST 191 CHARACTERS ON THE FIRST AVAILABLE ACTIVE LINE.
;
        INC ERRNBR             ;SET ERROR NUMBER TO 5602.
        MOV 0206,R0            ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
        JSR PC,WTWLCN          ;INITIALISE THE LINE CONTROL REGISTER.

```

6572	027036	012700	177670	MOV	#177670,R0	;PASS THE LPR CONTENTS.
6573	027042	004767	170534	JSR	PC,WTWLP	;SET THE LPR CONTENTS TO 38.4K BAUD.
6574	027046	012704	000012	MOV	#10.,R4	;PASS DELAY TIME OF 10 MILLI SECONDS.
6575	027052	004767	165204	JSR	PC,DELAY	;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6576	027056	010105		MOV	R1,R5	;COPY THE LINE NUMBER.
6577	027060	012702	002650	MOV	#BUF8AS,R2	;PASS THE START OF THE DATA PATTERN TO TX.
6578	027064	012703	000277	MOV	#191.,R3	;PASS THE LENGTH OF THE DATA PATTERN.
6579	027070	004767	165304	JSR	PC,DDDMA	;TRANSMIT THE DATA PATTERN.
6580	027074	103146		BCC	501	;EXIT IF ERROR FOUND DURING DMA TX.
6581						
6582						
6583						
6584						
6585	027076	005267	154750	INC	ERRNBR	;SET ERROR NUMBER TO 5603.
6586	027102	012701	170454	MOV	#170454,R1	;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6587	027106	016702	153114	MOV	CSRA,R2	;PASS THE ADDRESS OF THE CSR.
6588	027112	004767	170320	JSR	PC,WAIBIS	;WAIT FOR DMA TO COMPLETE, TX ACTION SET.
6589	027116	103135		BCC	501	;BRANCH IF FIFO EMPTY, ABORT THE TEST.
6590	027120	012704	000005	MOV	#5,R4	;PASS DELAY OF 5 MILLI SECS.
6591	027124	004767	165132	JSR	PC,DELAY	;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6592						
6593						
6594						
6595						
6596	027130	005267	154716	INC	ERRNBR	;SET ERROR NUMBER TO 5604.
6597	027134	010501		MOV	R5,R1	;PASS THE LINE NUMBER.
6598	027136	012702	002650	MOV	#BUF8AS,R2	;PASS THE START OF THE DATA PATTERN TO TX.
6599	027142	012703	000001	MOV	#1,R3	;PASS THE NUMBER OF
6600	027146	004767	165226	JSR	PC,DDDMA	;TX A NULL CHARACTER TO CAUSE AN XOFF.
6601	027152	103117		BCC	501	;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6602						
6603						
6604						
6605	027154	005267	154672	INC	ERRNBR	;SET ERROR NUMBER TO 5605.
6606	027160	012701	170012	MOV	#170012,R1	;PASS TIME-OUT VALUE OF 10 MILLI SECS.
6607	027164	016702	153036	MOV	CSRA,R2	;PASS THE ADDRESS OF THE CSR.
6608	027170	004767	170242	JSR	PC,WAIBIS	;WAIT FOR DMA TO COMPLETE, TX ACTION SET.
6609	027174	103106		BCC	501	;IF NO TX ACTION WAS RECEIVED, ABORT THE TEST.
6610	027176	012704	000005	MOV	#5,R4	;PASS DELAY OF 5 MILLI SECS.
6611	027202	004767	165054	JSR	PC,DELAY	;WAIT FOR XOFF TO GET INTO THE FIFO.
6612						
6613	027206	010577	153014	MOV	R5,BCSRA	;SELECT THE LINE READY FOR TRANSMISSION.
6614						
6615						
6616						
6617						
6618						
6619						
6620	027212	005005		CLR	R5	;CLEAR THE TX FLAG.
6621	027214	005004		CLR	R4	;CLEAR THE CHARACTER COUNTER.
6622	027216	012703	000300	MOV	#192.,R3	;SET UP READ COUNTER FOR THE FIRST 192 CHARS.
6623						
6624	027222	012700	000003	MOV	#3,R0	;SET READ COUNTER.
6625	027226	012701	170005	MOV	#170005,R1	;INDICATE TO TEST DATA VALID BIT, TIME OUT SMS.
6626	027232	016702	152772	MOV	RBUF A,R2	;INDICATE TO CHECK RECEIVE BUFFER REGISTER.
6627	027236	004767	170174	JSR	PC,WAIBIS	;WAIT FOR RECEIVED CHAR OR TIME OUT.
6628	027242	103046		BCC	141	;EXIT LOOP IF TIME-OUT, FIFO EMPTY.

C14

GZDMVBO DBU:11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 116-2

SEQ 171

```

6629 027244 005300          DEC R0          ;DECREMENT READ COUNTER.
6630 027246 005303          DEC R3          ;DECREMENT CHAR COUNTER.
6631 027250 003002          BGT 81          ;SKIP DISBL'G TX IF FIRST 192 CHARS NOT READ.
6632 027252 052705 100000   BIS #BIT15,R5   ;DISABLE ANY FURTHER TRANSMISSIONS.
6633
6634          ;
6635          ; CHECK IF THE READ CHARACTER IS A BMP CODE.
6636          ; IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
6637          ; ABORT THE TEST.
6638 027256 012767 012746 154566 81:  MOV #5606,,ERRNBR ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
6639 027264 004767 164506      JSR PC,CHKBMP ;CHECK IF CHARACTER IS A BMP CODE.
6640 027270 103446          BCS 161         ;GO REPORT ERROR AND ABORT TEST IF BMP FOUND.
6641
6642          ;
6643          ; CHECK FOR XOFF CHARACTER. IF ONE IS FOUND, COUNT IT.
6644          ; TRANSMIT A NULL CHARACTER UNTIL THE FIRST 192 CHARS HAVE BEEN READ.
6645 027272 122702 000023 101:  CMPB #23,R2      ;CHECK IF THE RECEIVED CHARACTER WAS AN XOFF.
6646 027276 001001          BNE 121         ;BRANCH IF CHARACTER WAS NOT AN XOFF.
6647 027300 005204          INC R4          ;INCREMENT THE XOFF CHAR FOUND COUNTER.
6648
6649 027302 005700          121:  TST R0          ;CHECK READ COUNT, TO SEE IF A CHAR CAN BE TX.
6650 027304 001350          BNE 61          ;BRANCH IF 3 CHARS HAVE NOT YET BEEN READ.
6651 027306 005705          TST R5          ;CHECK THE TRANSMISSION ENABLED FLAG.
6652 027310 100744          BMI 41          ;SKIP TRANSMITTING A CHARACTER IF TX DISABLED.
6653 027312 112777 000000 152714  MOVB #0,BFDATA ;TX A NULL CHARACTER.
6654 027320 010446          MOV R4,(-SP)    ;SAVE THE XOFF COUNT ON THE STACK.
6655
6656          ;
6657          ; WAIT FOR THE CHARACTER TO BE RECEIVED BEFORE CONTINUING THE TEST.
6658 027322 005267 154524      INC ERRNBR      ;SET ERROR NUMBER TO 5607.
6659 027326 012701 170012      MOV #170012,R1 ;PASS TIME-OUT VALUE OF 10 MILLI SECS.
6660 027332 016702 152670      MOV CSRA,R2   ;PASS THE ADDRESS OF THE CSR.
6661 027336 004767 170074      JSR PC,WAIBIS ;WAIT FOR DMA TO COMPLETE, TX ACTION SET.
6662 027342 103023          BCC 501         ;IF NO TX ACTION WAS RECEIVED, ABORT THE TEST.
6663 027344 012704 000005      MOV #5,R4     ;PASS DELAY OF 5 MILLI SECS.
6664 027350 004767 164706      JSR PC,DELAY  ;WAIT FOR XOFF TO GET INTO THE FIFO.
6665 027354 012604          MOV (SP)+,R4    ;RESTORE THE XOFF COUNT.
6666 027356 000721          BR 41           ;GO RESET THE READ COUNT AND GET NEXT CHAR.
6667
6668          ;
6669          ; CHECK IF THE CORRECT NUMBER OF XOFF'S WERE FOUND IN THE FIFO
6670          ; REPORT ERROR IF COUNT IS INCORRECT.
6671
6672 027360 012767 012750 154464 141:  MOV #5608,,ERRNBR ;SET ERROR NUMBER TO 5608.
6673 027366 020427 000077      CMP R4,#63    ;COMPARE THE EXPECTED AND ACTUAL XOFF COUNTS.
6674 027372 001411          BEQ 601         ;EXIT TEST IF SUCCESS.
6675 027374 012767 011762 154454  MOV #ER0503,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6676 027402 012701 006520      MOV #EM5402,R1 ;PASS THE MESSAGE TO BE REPORTED.
6677          ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
6678 027406          151:  ERROR          ;
6679 027410 000402          BR 601         ;EXIT THIS TEST.
6680
6681 027412 004767 167254 501:  JSR PC,TSABRT ;REPORT TEST ABORTED. ERROR # INDICATES FAULT.
6682 027416 005067 152626 601:  CLR CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6683
6684 027422          ENDTST

```

D14

C:\DHVBO.DHW.11 FUNC TST PAB12 MACRO M1200 15-MAR-84 09:28 PAGE 116-3
HARDWARE TEST F13QAI

SEQ 172

027422
027422 104401

L10041: TRAP C\$ETST

```

6686 .SBTTL  HARDWARE TEST          - FIMAVL -
6687 ;*****
6688 ;
6689 ;
6690 ;
6691 ;   THIS TEST CHECKS THAT THE DUT'S FIFO HALF LEVEL ALARM SYSTEM
6692 ;   BECOMES ACTIVE AND INACTIVE AT THE CORRECT LEVELS.
6693 ;   ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
6694 ;   HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
6695 ;   REPORTED LATER.
6696 ;   THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
6697 ;   INTERNAL LOOPBACK MODE.
6698 ;*****
6699 ;
6700 027424      BGNTST
6701 027424      SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.      T13:
6702 027424 012700 000240      MOV      #PRI05,R0
6703 027430 104441      TRAP      C:SPRI
6704 027432 000015      INUM ** INUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6705 027432 012767 000015 152614      MOV      #INUM,TSTNUM      ;SET UP THE TEST NUMBER.      (57)
6706 027440 012767 177777 152602      MOV      #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
6707 027446 012767 000001 154374      MOV      #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6708 027454 012767 013105 154370      MOV      #5701,ERRNBR      ;SET ERROR NUMBER TO 5701.
6709 027462 012767 006702 154364      MOV      #EM5701,ERRMSG      ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
6710 027470 012767 011762 154360      MOV      #ER0503,ERRBLK      ;SELECT THE ERROR REPORTING ROUTINE.
6711 ;
6712 ; RESET THE DUT TO A KNOWN STATE. REMOVE THE STATUS CODES FROM THE FIFO.
6713 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6714 ; THIS SUBROUTINE REPORTS ERROR >>>> 5701 <<<<.
6715 ;
6716 027476 004767 164514      JSR      PC,CLNRST      ;RESET THE DHU 11. REPORT ANY ERRORS FOUND.
6717 027502 103402      BCS      2$      ;SKIP EXITING TEST A SUCCESSFUL RESET.
6718 027504 000167 000364      JMP      60$      ;EXIT THIS TEST.
6719 2$:
6720 ;
6721 ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
6722 ;
6723 027510 004767 164754      JSR      PC,FINACT      ;FIND AN ACTIVE LINE.
6724 027514 103167      BCC      60$      ;EXIT IF NO ACTIVE LINES AVAILABLE.
6725 ;
6726 ; INITIALIZE THE 256 BYTE DATA PATTERN.
6727 ; ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
6728 ; NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
6729 ;
6730 027516 004767 165056      JSR      PC,INDTPX      ;INITIALISE THE DATA PATTERN.
6731 ;
6732 ; FILL THE FIFO AND THE UART'S 3 CHAR BUFFER BY TRANSMITTING 225 CHARS
6733 ; (IE 225 + 34 XOFF'S). TRANSMIT DATA PATTERN USING DMA, ON A SINGLE CHANNEL
6734 ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
6735 ;
6736 ;
6737 ; SET INTERNAL LOOPBACK, ENABLE TAUTO AND RECEIVER ON THE SELECTED LINE.
6738 ; TRANSMIT THE 225 CHARACTERS ON THE FIRST AVAILABLE ACTIVE LINE.
6739 ;
6738 027522 005267 154324      INC      ERRNBR      ;SET ERROR NUMBER TO 5702.
6739 027526 004767 167014      JSR      PC,SETPAR      ;SET UP PARAMETERS FOR TRANSMISSION.

```

6740	027532	012700	000341	MOV	0225.,R0	;PASS LENGTH OF DATA PATTERN.
6741	027536	004767	167242	JSR	PC,TXDATP	;TRANSMIT DATA PATTERN.
6742	027542	103152		BCC	501	;EXIT IF ERROR FOUND DURING TX.
6743	027544	010105		MOV	R1,R5	;COPY THE LINE NUMBER.
6744						
6745						
6746						
6747						
6748	027546	005267	154300	INC	ERRNBR	;SET ERROR NUMBER TO 5703.
6749	027552	004767	167734	JSR	PC,WAITTX	;WAIT FOR TRANSMISSION TO COMPLETE.
6750	027556	103144		BCC	501	;GO REPORT ERROR IF TX FAILED TO COMPLETE.
6751						
6752						
6753						
6754						
6755						
6756	027560	005267	154266	INC	ERRNBR	;SET ERROR NUMBER TO 5704.
6757	027564	012700	000202	MOV	0130.,R0	;PASS THE NUMBER OF CHARS TO READ.
6758	027570	004767	166060	JSR	PC,READBX	;READ THE FIRST 130 CHARS FROM THE FIFO.
6759	027574	103135		BCC	501	;GO REPORT ERROR IF BMP CODE FOUND.
6760	027576	005267	154250	INC	ERRNBR	;SET ERROR NUMBER TO 5705.
6761	027602	005701		TST	R1	;CHECK IF AN XON WAS FOUND.
6762	027604	001125		BNE	301	;GO REPORT ERROR IF AN XON WAS FOUND.
6763						
6764						
6765						
6766						
6767	027606	010577	152414	MOV	R5,0CSRA	;SELECT THE LINE READY FOR TRANSMISSION.
6768	027612	112777	000000	MOVB	00,0F DATA	;TRANSMIT A NULL CHARACTER.
6769	027620	005267	154226	INC	ERRNBR	;SET ERROR NUMBER TO 5706.
6770	027624	004767	167662	JSR	PC,WAITTX	;WAIT FOR TX TO COMPLETE.
6771	027630	103117		BCC	501	;GO REPORT ERROR IF TX DID NOT COMPLETE.
6772						
6773						
6774						
6775	027632	005267	154214	INC	ERRNBR	;SET ERROR NUMBER TO 5707.
6776	027636	012700	000003	MOV	03,R0	;SET THE READ COUNT TO 3.
6777	027642	004767	166006	JSR	PC,READBX	;READ 3 CHARACTERS FROM THE FIFO.
6778	027646	103110		BCC	501	;GO REPORT ERROR IF FIFO EMPTY.
6779	027650	005267	154176	INC	ERRNBR	;SET ERROR NUMBER TO 5708.
6780	027654	005701		TST	R1	;CHECK IF AN XON WAS FOUND.
6781	027656	001102		BNE	401	;GO REPORT ERROR IF AN XON WAS FOUND.
6782						
6783						
6784						
6785	027660	012700	000076	MOV	062.,R0	;PASS LENGTH OF DATA PATTERN.
6786	027664	010501		MOV	R5,R1	;PASS THE LINE NUMBER.
6787	027666	005267	154160	INC	ERRNBR	;SET ERROR NUMBER TO 5709.
6788	027672	004767	167106	JSR	PC,TXDATP	;TRANSMIT DATA PATTERN.
6789	027676	103074		BCC	501	;EXIT IF ERROR FOUND DURING TX.
6790						
6791						
6792						
6793						
6794						
6795	027700	005267	154146	INC	ERRNBR	;SET ERROR NUMBER TO 5710.
6796	027704	004767	167602	JSR	PC,WAITTX	;WAIT FOR TX TO COMPLETE.

```

6797 027710 103067          BCC      50$      ;GO REPORT ERROR IF TX FAILED TO COMPLETE.
6798
6799          ; READ THE FIRST 126 CHARACTERS.
6800          ; READ THE NEXT 4 CHARACTERS AND CHECK IF THEY ARE IN THE FOLLOWING ORDER
6801          ; NULL, XOFF, XON, NULL.
6802
6803 027712 005267 154134      INC      ERRNBR      ;SET ERROR NUMBER TO 5711.
6804 027716 012700 000176      MOV      #126.,R0      ;SET UP READ COUNTER.
6805 027722 004767 165726      JSR      PC,READBX      ;READ THE FIRST 126 CHARS.
6806 027726 103060              BCC      50$      ;GO REPORT THE ERROR IF FIFO EMPTY.
6807 027730 005267 154116      INC      ERRNBR      ;SET ERROR NUMBER TO 5712.
6808 027734 005701              TST      R1      ;CHECK IF AN XON WAS FOUND.
6809 027736 001052              BNE      40$      ;GO REPORT ERROR IF AN XON WAS FOUND.
6810 027740 005267 154106      INC      ERRNBR      ;SET ERROR NUMBER TO 5713.
6811 027744 012701 006520      MOV      #EM5402,R1      ;PASS THE MESSAGE TO BE REPORTED.
6812 027750 016703 152254      MOV      RBUFA,R3      ;GET THE RECEIVER BUFFER ADDRESS.
6813 027754 011302              MOV      (R3),R2      ;READ THE NULL CHARACTER FROM THE FIFO.
6814 027756 120227 000000      CMPB     R2,#0000      ;CHECK IF IT IS A NULL CHARACTER.
6815 027762 001040              BNE      40$      ;GO REPORT THE ERROR IF NOT THE SAME.
6816 027764 005267 154062      INC      ERRNBR      ;SET ERROR NUMBER TO 5714.
6817 027770 011302              MOV      (R3),R2      ;READ THE XOFF FROM THE FIFO.
6818 027772 120227 000023      CMPB     R2,#23      ;CHECK IF THE READ CHAR IS AN XOFF.
6819 027776 001032              BNE      40$      ;GO REPORT THE ERROR IF NOT THE SAME.
6820 030000 011302              MOV      (R3),R2      ;READ THE XON FROM THE FIFO.
6821 030002 005267 154044      INC      ERRNBR      ;SET ERROR NUMBER TO 5715.
6822 030006 120227 000021      CMPB     R2,#21      ;CHECK IF THE READ CHARACTER IS AN XON.
6823 030012 001024              BNE      40$      ;GO REPORT THE ERROR IF NOT THE SAME.
6824 030014 005267 154032      INC      ERRNBR      ;SET ERROR NUMBER TO 5716.
6825 030020 011302              MOV      (R3),R2      ;READ THE NULL CHARACTER FROM THE FIFO.
6826 030022 120227 000000      CMPB     R2,#0000      ;CHECK IF IT IS A NULL CHARACTER.
6827 030026 001016              BNE      40$      ;GO REPORT THE ERROR IF NOT THE SAME.
6828
6829
6830          ; READ THE REMAINING CHARACTERS FROM THE FIFO.
6831
6832 030030 012700 000075      6$:      MOV      #61.,R0      ;SET UP READ COUNTER.
6833 030034 005267 154012      INC      ERRNBR      ;SET ERROR NUMBER TO 5717.
6834 030040 004767 165610      JSR      PC,READBX      ;READ THE FIRST 61 CHARS.
6835 030044 103011              BCC      50$      ;GO REPORT THE ERROR IF FIFO EMPTY.
6836 030046 005267 154000      INC      ERRNBR      ;SET ERROR NUMBER TO 5718.
6837 030052 005701              TST      R1      ;CHECK IF AN XON WAS FOUND.
6838 030054 001003              BNE      40$      ;GO REPORT ERROR IF AN XON WAS FOUND.
6839 030056 000406              BR       60$      ;EXIT THE TEST.
6840 030060 012701 006520      30$:      MOV      #EM5402,R1      ;SET UP THE MESSAGE
6841
6842 030064              40$:      ERROR      ; "FIFO ALARM SIGNAL DEFECTIVE".
6843 030066 000402              BR       60$      ;EXIT THE TEST.
6844
6845 030070 004767 166576      50$:      JSR      PC,ABRT      ;REPORT TEST ABORTED. ERROR # INDICATES FAULT.
6846 030074 005067 152150      60$:      CLR      CIALCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6847
6848 030100              ENDTST
6849 030100
6850 030100 104401              L10042:      TRAP      C$ETST

```

```

6850 .SBTTL  HARDWARE TEST          - RXTIMER -
6851 ;* *****
6852 ;*          - RXTIMER REG TEST -
6853 ;*      THIS TEST VERIFIES THAT THE RXTIMER DELAYS ANY RX-INTS BY THE
6854 ;*      REQUESTED AMOUNT AND THAT WHEN THE RXFIFO IS MORE THAN 3/4 FULL
6855 ;*      THE RXTIMER VALUE IS IGNORED AND AN INTERRUPT OCCURS IMMEDIATELY.
6856 ;*      DUE TO THE DIFFERENCES IN LTC HANDLING OF DIFFERENT VERSIONS OF
6857 ;*      THE DRS AND LTC AVAILABILITY ON DIFFERENT PDP-11 MACHINES THE
6858 ;*      RX-INT CAN ONLY BE TIMED TO WITHIN +/- 20% OF THE RXTIMER VALUE.
6859 ;*      THIS TEST IS PERFORMED IN INTERNAL LOOPBACK ON THE FIRST ACTIVE LINE.
6860 ;*
6861 ;* *****
6862
6863 030102      BGNTST
6864 030102      SETPRI 0PRI05          ;ALLOW LTC INTERRUPTS,      T14::
6865 030102      012700 000240          ;MOV      0PRI05,R0
6866 030106      104441          ;TRAP    C0SPRI
6867 030110      000016          ;TNUM == TNUM + 1      ;INCREMENT ASSEMBLY TIME TEST COUNTER
6868 030116      012767 000016 152136  ;MOV      0TNUM,TSTNUM      ;SET UP THE TEST NUMBER.
6869 030124      012767 177777 152124  ;MOV      0-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
6870 030132      012767 000001 153716  ;MOV      01,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6871 030140      012767 013251 153712  ;MOV      05801,ERRNBR    ;SET THE ERROR NUMBER TO 5801.
6872 030146      005067 152100          ;MOV      0EM5801,ERRMSG   ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
6873 ;*          CLR      EXOERR          ;CLEAR THE "EXIT ON ERROR" FLAG.
6874 ;*
6875 ;*      RESET THE DEVICE AND LEAVE THE SELFTEST CODES IN THE FIFO.
6876 ;*
6877 030152      004767 166124          ;JSR      PC,RESETT      ;RESET THE OUT.
6878 030156      103402          ;BCS      ,+6          ;CONTINUE IF FIFO PURGED
6879 030160      000167 000732          ;JMP      60$          ;REPORT THE RESET FAILURE.
6880 ;*
6881 ;*      SET UP THE INTERRUPT SERVICE ROUTINE THAT DETECTS THE RX-INT.
6882 ;*
6883 030164      SETVEC RXVECA,0RXDECT,0PRI06
6884 030164      012746 000300          ;MOV      0PRI06, -(SP)
6885 030170      012746 017702          ;MOV      0RXDECT, -(SP)
6886 030174      016746 152014          ;MOV      RXVECA, (SP)
6887 030200      012746 000003          ;MOV      03, -(SP)
6888 030204      104437          ;TRAP    C0SVEC
6889 030206      062706 000010          ;ADD      010,SP
6890 030212      SETPRI 0PRI04          ;ALLOW DEVICE INTERRUPTS,
6891 030212      012700 000200          ;MOV      0PRI04,R0
6892 030216      104441          ;TRAP    C0SPRI
6893 030220      005067 152042          ;CLR      RXINTC          ;CLEAR THE RX-INT COUNT.
6894 ;*
6895 ;*      ENABLE RX-INTS AND WAIT FOR ONE TO OCCUR.
6896 ;*
6897 030224      004767 166224          ;JSR      PC,RXIE1      ;ENABLE RX-INTS.
6898 030230      012704 000005          ;MOV      05,R4          ;SET THE DELAY OF 5 MILLI SECS.
6899 030234      004767 164022          ;JSR      PC,DELAY      ;DELAY WHILE THE INT OCCURS.
6900 030240      004767 166150          ;JSR      PC,RXIE0      ;DISABLE RX-INTS.
6901 030244      005267 153602          ;INC      ERRNBR          ;SET THE ERROR NUMBER TO 5802.
6902 030250      005767 152012          ;TST      RXINTC          ;TEST IF AN INTERRUPT OCCURED.
6903 030254      001002          ;BNE      ,+6          ;CONTINUE IF AN INTERRUPT OCCURED.
6904 030256      000167 000630          ;JMP      50$          ;REPORT THE ERROR IF NO INTERRUPT.
6905 ;*

```

```

6896 ; SET INTERNAL LOOPBACK ON THE FIRST ACTIVE LINE AND ENABLE RECIEVERS. SET UP
6897 ; THE LPR'S TO 38.4K BAUD, 8 BITS/CHARA, ODD PARITY, 2 STOP BITS.
6898 ; -
6899 030262 004767 164202 JSR PC,FINACT ;FIND AN ACTIVE LINE FOR THIS TEST.
6900 030266 103402 BCS .+6 ;CONTINUE IF A LINE HAS BEEN FOUND.
6901 030270 000167 000622 JMP 60$ ;EXIT THE TEST IF NO LINES ACTIVE.
6902 030274 012700 000204 MOV $204,R0 ;PASS PARAMETER FOR INTERNAL LOOFBACK,
6903 ;ENABLE RECIEVERS.
6904 030300 004767 167246 JSR PC,WTWLNC ;INITIALISE THE LINE CONTROL REGS.
6905 030304 012700 177670 MOV $177670,R0 ;PASS THE LPR CONTENTS.
6906 030310 004767 167266 JSR PC,WTWLPR ;SET THE LPR'S TO 38.4K BAUD.
6907 030314 012704 000012 MOV $10,R4 ;PASS DELAY TIME OF 10 MILLI SECS.
6908 030320 004767 163736 JSR PC,DELAY ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6909 ;
6910 ; SET UP THE LOOP TO TEST THE RXTIMER WITH DELAYS OF 15,31,63,127 AND 255 MS.
6911 ; DMA 191 CHARACTERS INTO THE FIFO AND THEN ENABLE INTERRUPTS, VERIFY THAT
6912 ; THE INTERRUPT OCCURS WITHIN +/- 20% OF THE RXTIMER VALUE.
6913 ; -
6914 030324 010167 000616 MOV R1,70$ ;SAVE THE LINE NUMBER.
6915 030330 012705 000020 MOV $16,R5 ;SET THE FIRST (RXTIMER VALUE + 1).
6916 030334 012767 013253 153510 2$; MOV $5803,ERRNBR ;SET THE ERROR NUMBER TO 5803.
6917 030342 004767 165022 JSR PC,PUFIFO ;PURGE THE RXFIFO.
6918 030346 103402 BCS .+6 ;CONTINUE IF SUCCESSFUL.
6919 030350 000167 000536 JMP 50$ ;REPORT THE ERROR IF FIFO FAILED TO PURGE.
6920 030354 016701 000566 MOV 70$,R1 ;PASS THE LINE NUMBER.
6921 030360 012703 000277 MOV $191,R3 ;PASS THE NUMBER OF CHARS TO DMA.
6922 030364 005267 153462 INC ERRNBR ;SET THE ERROR NUMBER TO 5804.
6923 030370 004767 163726 JSR PC,DMABUF ;PERFORM THE DMA FROM ADDR $BUFBA5,THIS SUBR
6924 ;PRODUCES ERRORS >>>> 5804 THRU 5805 <<<<<.
6925 030374 103402 BCS .+6 ;CONTINUE IF SUCCESSFUL.
6926 030376 000167 000510 JMP 50$ ;REPORT THE ERPOR IF ONE OCCURED.
6927 ;
6928 ; CALCULATE THE TIME-OUT VALUE FOR THE RX-INT, SET UP THE RXTIMER, AND
6929 ; WAIT FOR THE RX-INT.
6930 ; -
6931 030402 010501 MOV R5,R1 ;COPY THE RXTIMER VALUE + 1.
6932 030404 006301 ASL R1 ;MULTIPLY BY 4 TO OBTAIN,
6933 030406 006301 ASL R1 ;THE TIME-OUT FOR THE RX-INT.
6934 030410 105077 151612 CLR R0 ;CLEAR THE IND.ADDR.REG BITS OF THE CSR READY,
6935 ;FOR THE WRITE TO THE RXTIMER REG.
6936 030414 010500 MOV R5,R0 ;COPY THE RXTIMER VALUE +1.
6937 030416 005300 DEC R0 ;GET THE RXTIMER VALUE
6938 030420 110077 151604 MOV R0,$RXTMA ;LOAD THE RXTIMER REG.
6939 030424 012704 000002 MOV $2,R4 ;SET DELAY OF 2 MS.
6940 030430 004767 163626 JSR PC,DELAY ;DELAY TO ALLOW THE RXTIMER VALUE TO UPDATE.
6941 030434 012702 000001 MOV $BIT0,R2 ;INDICATE TO TEST BIT0.
6942 030440 010203 MOV R2,R3 ;INDICATE TO TEST FOR A "1".
6943 030442 012704 002266 MOV $RXINTC,R4 ;PASS ADDR OF WORD TO TEST.
6944 030446 005067 151614 CLR RXINTC ;CLEAR THE RX INT COUNT.
6945 030452 012767 000100 151576 MOV $BIT06,IESTAT ;SET THE RX-INT-ENBL BIT IN IESTAT.
6946 030460 016777 151572 151540 MOV IESTAT,$CSRA ;ENABLE RX INTS.
6947 030466 004767 164236 JSR PC,MSLGET ;WAIT FOR THE INT TO OCCUR.
6948 030472 103415 BCS 4$ ;AVOID ERROR REPORT IF THE INTERRUPT OCCURED.
6949 ;
6950 ; REPORT THE TIME OUT ERROR. >>>> 5806 <<<<<.
6951 ; -
6952 030474 010502 MOV R5,R2 ;PASS THE RXTIMER VALUE TO,

```

```

6953 030476 005302          DEC    R2          ;THE ERROR REPORTING ROUTINE.
6954 030500 012701 007265    MOV    @EM5805,R1    ;PASS THE MESSAGE.
6955                                ; "RXTIMER BAD, TIME-OUT OCCURED WAITING FOR
6956                                ; THE RX-INT".
6957
6958 030504          ERRDF    5806,EM5801,ER5801 ;REPORT ERROR 5806.
        030504 104455
        030506 013256
        030510 006757
        030512 012112
                                TRAP    C1ERRDF
                                .WORD    5806
                                .WORD    EM5801
                                .WORD    ER5801
6959
6960 030514 032767 000100 151466    BIT    @BIT06,OPTION ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
6961 030522 001575          BEQ    60$          ;EXIT THE TEST IF IT HASN'T.
6962 030524 000454          BR     6$           ;BRANCH TO TEST ANOTHER RXTIMER VALUE.
6963
        ; *
        ; CHECK THAT THE INTERRUPT OCCURED WITHIN +/- 20% OF THE RXTIMER VALUE.
        ; THIS SUBROUTINE REPORTS ERROR >>>>> 5807 <<<<<.
        ; -
6964
6965
6966
6967 030526 012767 013257 153316 4$:    MOV    @5807,ERRNBR ;SET THE ERROR NUMBER TO 5807.
6968 030534 010502          MOV    R5,R2          ;PASS THE RXTIMER VALUE TO,
6969 030536 005302          DEC    R2              ;THE "CHECK TIME" SUBR.
6970 030540 004767 163302    JSR    PC,CKRXTM    ;CHECK THE TIME TAKEN AND REPORT ANY ERROR.
6971 030544 005767 151502    TST    EXOERR      ;TEST THE "EXIT ON ERROR" FLAG.
6972 030550 001162          BNE    60$          ;EXIT IF SET.
6973 030552 004767 165636    JSR    PC,RXIE0    ;DISABLE RX-INTS.
6974
        ; *
        ; DMA ANOTHER CHARACTER TO FILL THE FIFO TO THE 75% LEVEL, AND CHECK THAT THE
        ; RX-INT OCCURS IMMEDIATELY.
        ; -
6975
6976
6977
6978 030556 016701 000364          MOV    70$,R1          ;PASS THE LINE NUMBER.
6979 030562 012703 000001          MOV    @1,R3          ;PASS THE NUMBER OF CHARS TO DMA.
6980 030566 005267 153260          INC    ERRNBR        ;SET THE ERROR NUMBER TO 5808.
6981 030572 004767 163524          JSR    PC,DMABUF      ;PERFORM THE DMA FROM ADDR @8UF8AS,THIS SUBR
6982                                ;PRODUCEJ ERRORS >>>>> 5808 THRU 5809 <<<<<.
6983 030576 103145          BCC    50$          ;REPORT THE ERROR IF ONE OCCURED.
6984 030600 005067 151462          CLR    RXINTC        ;CLEAR THE RX-INT COUNT.
6985 030604 012701 000005          MOV    @5,R1          ;SET THE TIME-OUT TO 5 MS.
6986 030610 012702 002266          MOV    @RXINTC,R2     ;PASS ADDR OF WORD TO TEST.
6987 030614 004767 165634          JSR    PC,RXIE1        ;ENABLE INTERRUPTS.
6988 030620 004767 166612          JSR    PC,WAIBIS       ;WAIT FOR THE INT TO OCCUR.
6989 030624 103414          BCS    6$           ;AVOID THE ERROR IF AN INTERRUPT OCCURED.
6990
        ; *
        ; REPORT THE ERROR, RX-INT DID NOT OCCUR IMMEDIATELY.>>>>> 5810 <<<<<.
        ; -
6991
6992
6993 030626 010502          MOV    15,R2          ;PASS THE RXTIMER VALUE.
6994 030630 005302          DEC    R2
6995 030632 012701 007077          MOV    @EM5803,R1    ;PASS THE MESSAGE.
6996                                ; "RXTIMER BAD, RX-INT DID NOT OCCUR
6997                                ; IMMEDIATELY WHEN RXFIFO 3/4 FULL".
6998
6999 030636          ERRDF    5810,EM5801,ER5801 ; REPORT ERROR 5810.
        030636 104455
        030640 013262
        030642 006757
        030644 012112
                                TRAP    C1ERRDF
                                .WORD    5810
                                .WORD    EM5801
                                .WORD    ER5801
7000
7001 030646 032767 000100 151334    BIT    @BIT06,OPTION ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?

```

```

7002 030654 001520          BEQ      60$          ;EXIT THE TEST IF IT HASN'T.
7003
7004          ;*
7005          ; SELECT ANOTHER VALUE FOR THE RXTIMER OR IF ALL VALUES HAVE BEEN TESTED THEN
7006          ; TEST THE RXTIMER WITH INDEFINATE DELAY SET.
7007 030656 004767 165532    6$:   JSR      PC,RXIE0      ;DISABLE INTERRUPTS.
7008 030662 006305          ASL      R5              ;MULTIPLY (RXTIMER VALUE + 1) BY 2.
7009 030664 020527 000400    CMP      R5,#256.        ;HAVE ALL VALUES BEEN TESTED ?
7010 030670 003621          BLE      2$              ;BRANCH AND TEST ANOTHER VALUE IF NOT.
7011
7012          ;*
7013          ; VERIFY THAT WHEN RXTIMER VALUE IS 0 THE INTERRUPT IS DELAYED INDEFINITELY,
7014          ; UNLESS THE RXFIFO IS 75% FULL OR MORE.
7015 030672 012767 013263 153152    MOV      #5811.,ERRNBR ;SET THE ERROR NUMBER TO 5811.
7016 030700 004767 164464    JSR      PC,PURFIFO    ;PURGE THE RXFIFO.
7017 030704 103102          BCC      50$          ;REPORT THE ERROR IF THE FIFO FAILED TO PURGE.
7018 030706 016701 000234    MOV      70$,R1        ;PASS THE LINE NUMBER.
7019 030712 012703 000277    MOV      #191.,R3       ;PASS THE NUMBER OF CHARS TO DMA.
7020 030716 005267 153130    INC      ERRNBR        ;SET THE ERROR NUMBER TO 5812.
7021 030722 004767 163374    JSR      PC,DMABUF    ;PERFORM THE DMA FROM ADDR #BUFBAS,THIS SUBR
7022                                ;PRODUCES ERRORS >>>> 5812 THRU 5813 <<<<.
7023 030726 103071          BCC      50$          ;REPORT THE ERROR IF ONE OCCURED.
7024 030730 012701 001750    MOV      #1750,R1      ;INDICATE TO TEST BIT0 WITH TIME OUT OF 1 SEC.
7025 030734 012702 002266    MOV      #RXINTC,R2     ;PASS THE ADDR OF THE WORD TO TEST.
7026 030740 005067 151322    CLR      RXINTC       ;CLEAR THE RX-INT COUNT.
7027 030744 105077 151256    CLRB     #CSRA        ;CLEAR THE IND.ADDR.REG BITS OF THE CSR READY,
7028                                ;FOR THE WRITE TO THE RXTIMER REG.
7029 030750 105077 151254    CLRB     #RXTMA        ;SET THE VALUE 0 IN THE RXTIMER.
7030 030754 012704 000002    MOV      #2,R4         ;SET THE DELAY OF 2 MS.
7031 030760 004767 163276    JSR      PC,DELAY    ;DELAY TO ALLOW THE RXTIMER VALUE TO UPDATE.
7032 030764 004767 165464    JSR      PC,RXIE1      ;ENABLE RX-INTS.
7033 030770 004767 166442    JSR      PC,WAIBIS     ;WAIT FOR THE INTERRUPT TO OCCUR.
7034 030774 103007          BCC      8$              ;AVOID THE ERROR IF NO INTERRUPT.
7035
7036          ;*
7037          ; REPORT THE ERROR, RX-INT OCCURED WITH RXTIMER VALUE ZERO.>>>> 5814 <<<<.
7038 030776 012701 007201    8$:   MOV      #EM5804,R1      ;PASS THE MESSAGE,
7039                                ; "RXTIMER BAD, RX-INT OCCURED WITH RXTIMER
7040                                ; VALUE ZERO".
7041
7042          ERRDF      5814,EM5801,ER0503 ; REPORT ERROR 5814.
7043
7044          TRAP      C:ERDF
7045          .WORD      5814
7046          .WORD      EM5801
7047          .WORD      ER0503
7048 031012 000441          BR      60$          ;EXIT THE TEST.
7049
7050          ;*
7051          ; VERIFY THAT WHEN THE FIFO IS 75% FULL THE INTERRUPT OCCURS IMMEDIATELY.
7052 031014 004767 165374    8$:   JSR      PC,RXIE0      ;DISABLE RX-INTS.
7053 031020 012767 013267 153024    MOV      #5815.,ERRNBR ;SET THE ERROR NUMBER TO 5815.
7054 031026 016701 000114    MOV      70$,R1        ;PASS THE LINE NUMBER.
7055 031032 012703 000001    MOV      #1,R3         ;PASS THE NUMBER OF CHARS TO DMA.
7056 031036 004767 163260    JSR      PC,DMABUF    ;PERFORM THE DMA FROM ADDR #BUFBAS,THIS SUBR
7057                                ;PRODUCES ERRORS >>>> 5815 THRU 5816 <<<<.
7058          BCC      50$          ;REPORT THE ERROR IF ONE OCCURED.

```

```

7055 031044 005067 151216          CLR    RXINTC          ;CLEAR THE RX-INT COUNT.
7056 031050 012701 000005          MOV    #5,R1          ;SET THE TIME-OUT TO 5 MS.
7057 031054 012702 002266          MOV    @RXINTC,R2        ;PASS ADDR OF WORD TO TEST.
7058 031060 004767 165370          JSR    PC,RXIE1          ;ENABLE INTERRUPTS.
7059 031064 004767 166346          JSR    PC,WAIBIS          ;WAIT FOR THE INT TO OCCUR.
7060 031070 103412                  BCS     60$              ;EXIT THE TEST IF AN INTERRUPT OCCURED.
7061                                     ;+
7062                                     ; REPORT THE ERROR, RX-INT DID NOT OCCUR IMMEDIATELY.>>>> 5817 <<<<.
7063                                     ;-
7064 031072 005002                  CLR    R2              ;PASS THE RXTIMER VALUE.
7065 031074 012701 007077          MOV    @EM5803,R1          ;PASS THE MESSAGE.
7066                                     ERRDF  5817,EM5801,ER5801 ; REPORT ERROR 5817.
7067 031100                  TRAP    C$ERRDF
       031100 104455                  .WORD 5817
       031102 013271                  .WORD EM5801
       031104 006757                  .WORD ER5801
       031106 012112
7068
7069 031110 000402                  BR      60$              ;EXIT THE TEST.
7070
7071 031112 004767 165554          50$: JSR    PC,ISABRT          ;REPORT NON-RELATED TEST ERROR.
7072 031116 012700 000340          60$: SETPRI @PRI07          ;DISABLE ALL INTERRUPTS.
       031116 012700 000340                  MOV    @PRI07,R0
       031122 104441                  TRAP    C$SPRI
7073 031124 004767 165264          JSR    PC,RXIE0          ;DISABLE DEVICE RX-INTS.
7074 031130 016700 151060          CLRVEC RXVECA          ;CLEAR DOWN THE RX VECTOR.
       031130 016700 151060                  MOV    RXVECA,R0
       031134 104436                  TRAP    C$CVEC
7075 031136 005067 151106          CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7076 031142 012700 000340          EXIT TST
       031142 104432                  TRAP    C$EXIT
       031144 000004                  .WORD L10043-
7077 031146 000000          70$: .WORD 0              ;LOCAL STORAGE FOR LINE NUMBER USED IN THE TEST.
7078 031150                  L10043:
       031150 104401                  TRAP    C$ETST

```

```

7080 .SBTTL  HARDWARE TEST - TXACTF -
7081 ;+ *****
7082 ;+ TX ACTION FIFO TEST -
7083 ;+
7084 ;+ THIS TEST VERIFIES THAT THE DUT'S TX-ACTION FIFO CAN CORRECTLY
7085 ;+ HOLD 16 TX-ACTIONS. ONE CHARACTER IS TRANSMITTED ON EACH LINE
7086 ;+ USING DMA, THE TX-ACTIONS ARE THEN READ FROM THE FIFO, VERIFYING
7087 ;+ THAT THEY ARE IN THE CORRECT ORDER AND THAT THERE ARE 16 OF THEM.
7088 ;+ THE TEST ALSO VERIFIES THAT THE DUT WILL NOT SEND TX-INTS AFTER
7089 ;+ THE TX ACTION FIFO HAS BEEN EMPTIED.
7090 ;+ THIS TEST IS PERFORMED IN INTERNAL LOOPBACK ON ALL LINES.
7091 ;+
7092 ;+ *****
7093
7094 031152 BGNTST
7095 031152 T15::
7096 031152 SETPRI 0PRI05 ;ALLOW LTC INTERRUPTS.
7097 031152 012700 000240 MOV 0PRI05,R0
7098 031152 104441 TRAP C$SPRI
7099 000017 TNUM == TNUM + 1 ;INCREMENT ASSEMBLY TIME TEST COUNTER
7100 031160 012767 000017 151066 MOV 0TNUM,STNUM ;SET UP THE TEST NUMBER.
7101 031166 012767 177777 151054 MOV 0-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
7102 031174 012767 000001 152646 MOV 01,ERRYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
7103 031202 012767 013415 152642 MOV 05901,ERRNBR ;SET THE ERROR NUMBER TO 5901.
7104 031210 012767 007346 152636 MOV 0EM5901,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
7105 031216 012767 012170 152632 MOV 0ER5901,ERRBLK ;SET THE ERROR REPORTING ROUTINE.
7106
7107 ;+
7108 ;+ RESET THE DUT TO A KNOWN STATE, REMOVE ANY STATUS CODES IN THE FIFO.
7109 ;+ CLEAR THE RX AND TX ENABLE BITS IN THE CSR.
7110 ;+ THIS SUBROUTINE REPORTS ERROR >>>> 5901 <<<<.
7111 ;+
7112 031224 004767 162766 JSR PC,CLNRST ;RESET THE DHU-11 REPORT ANY ERRORS FOUND.
7113 031230 103402 BCS .+6 ;SKIP EXIT TEST IF NO FATAL ERROR FOUND.
7114 031232 000167 000474 JMP 60$ ;EXIT THE TEST IF FATAL ERROR FOUND.
7115
7116 ;+
7117 ;+ SET INTERNAL LOOPBACK 0 ALL LINES AND ENABLE RECIEVERS. SET UP THE
7118 ;+ LPR'S TO 38.4K BAUD, 8 BITS/CHARA, ODD PARITY, 2 STOP BITS.
7119 ;+
7120 031236 012705 177777 MOV 0MAPLNS,R5 ;INDICATE TO SET UP ALL LINES.
7121 031242 012700 00020 MOV 0204,R0 ;PASS PARAMETER FOR INTERNAL LOOPBACK,
7122 031246 004767 166300 JSR PC,WTWLN ;INITIALISE THE LINE CONTROL REGS.
7123 031252 012700 177670 MOV 017070,R0 ;PASS THE LPR CONTENTS.
7124 031256 004767 166320 JSR PC,WTWLP ;SET THE LPR'S TO 38.4K BAUD.
7125 031262 012704 000012 MOV 010,R4 ;PASS DELAY TIME OF 10 MILLI SECS.
7126 031266 004767 162770 JSR PC,DELAY ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
7127
7128 ;+
7129 ;+ INITIATE A DMA ON EACH LINE AND WAIT FOR ALL DMA'S TO COMPLETE.
7130 ;+
7131 031272 005001 CLR R1 ;PASS THE FIRST LINE NUMBER.
7132 031274 012702 002650 MOV 0BUFBAS,R2 ;PASS THE START OF THE DATA PATTERN TO TX.
7133 031300 012703 000001 MOV 01,R3 ;PASS THE LENGTH OF THE DATA PATTERN.

```

```

7134 03130 012704 000005      MOV    #5,R4      ;PASS THE DELAY TIME OF 5 MILLI SECS.
7135 03131 005267 152536      INC    ERRNBR      ;SET THE ERROR NUMBER TO 5902.
7136 031314 004767 163060      JSR    PC,DODMA      ;TRANSMIT THE DATA PATTERN.
7137 031320 103402      BCS    .+6      ;CONTINUE IF SUCCESSFUL.
7138 031322 000167 000400      JMP    50$      ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
7139
7140      ;+
7141      ; WAIT FOR THE DMA TO COMPLEATE BEFORE INITIATING ANOTHER.
7142 031326 004767 162730      JSR    PC,DELAY      ;WAIT 5 MILLI SECS FOR THE DMA TO COMPLEATE.
7143 031332 005201      INC    R1      ;INCREMENT THE LINE NUMBER.
7144 031334 022701 000020      CMP    #16.,R1      ;BRANCH TO INITIATE ANOTHER DMA IF
7145 031340 001365      BNE    2$      ;NOT ALL LINES SERVED.
7146
7147      ;+
7148      ; READ THE TX-ACTIONS FROM THE TX_ACTION FIFO AND VERIFY THAT THEY OCCURED
7149      ; IN THE CORRECT ORDER.
7150 031342 005267 152504      INC    ERRNBR      ;SET THE ERROR NUMBER TO 5903.
7151 031346 012703 007402      MOV    #EM5902,R3      ;SET THE ERROR MESSAGE TO,
7152
7153 031352 005001      CLR    R1      ;"TX-ACTION RECIEVED FROM THE WRONG LINE".
7154 031354 017702 150646      MOV    @CSRA,R2      ;CLEAR THE LINE NUMBER.
7155 031360 100150      BPL    14$      ;READ THE CSR.
7156 031362 000302      SWAB   R2      ;BRANCH TO REPORT ERROR >>5904<<, IF NO TX-ACT.
7157 031364 042702 177760      BIC    #177760,R2      ;PUT THE "X LINE NUMBER IN THE LOW BYTE.
7158 031370 020201      CMP    R2,R1      ;CLEAR THE CLUTTER FROM THE LINE NUMBER.
7159
7160 031372 001405      BEQ    6$      ;COMPARE THE ACTUAL LINE NUMBER OF THE
7161 031374      ERROR      ;TX-ACTION WITH THE EXPECTED NUMBER.
7162 031376 104460      TRAP    C$ERROR      ;SKIP THE ERROR REPORT IF CORRECT.
7163 031404 001552      BIT    #BIT06,OPTION      ;REPORT THE ERROR >>>> 5903 <<<<.
7164 031406 005201      BEQ    60$      ;EXIT IF EXTENDED ERROR REPORTING HAS NOT
7165 031410 022701 000020      INC    R1      ;BEEN REQUESTED.
7166 031414 001357      CMP    #16.,R1      ;INCREMENT THE EXPECTED LINE NUMBER.
7167
7168      ;+
7169      ; 16 TX-ACTIONS HAVE BEEN READ, THE TX-ACTION BIT SHOULD NOW BE CLEAR.
7170      ; CHECK THAT IT IS CLEAR, IF IT ISN'T THEN COUNT THE NUMBER OF EXTRA
7171      ; TX-ACTIONS RECIEVED AND REPORT THE ERROR.
7172
7173 031416 005777 150604      IST    @CSRA      ;READ THE CSR.
7174 031422 100024      BPL    10$      ;BRANCH IF THE TX-ACTION BIT IS CLEAR, TO
7175
7176 031424 012767 013421 152420      MOV    #5905.,ERRNBR      ;TEST THE TX-INTERRUPTS.
7177 031432 012702 000021      MOV    #17.,R2      ;SET THE ERROR NUMBER TO 5905.
7178 031436 005777 150564      IST    @CSRA      ;SET R2 TO BE THE NUMBER OF TX-ACTIONS FOUND.
7179 031442 100123      BPL    16$      ;READ THE CSR.
7180
7181 031444 005202      INC    R2      ;BRANCH AND REPORT ERROR IF THE TX-ACTION FIFO,
7182 031446 022702 000145      CMP    #101.,R2      ;FINALLY CLEARED.
7183 031452 001371      BNE    8$      ;INCREMENT THE NUMBER OF TX-ACTIONS FOUND.
7184
7185      ;+
7186      ; REPORT THE ERROR "TX-ACTION FIFO WOULD NOT EMPTY"
7187
7188 031454 005267 152372      INC    ERRNBR      ;SET THE ERROR NUMBER TO 5906.
7189 031460 012701 007570      MOV    #EM5904,R1      ;SET THE ERROR MESSAGE TO,

```

815

CZDHVRO DMU 11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 119-2
HARDWARE TEST TXACTF

SEQ 183

CZDH
HARD

```

7190
7191 031464 012767 011762 152364      MOV    #ERO503,ERRBLK    ; "TX ACTION FIFO WOULD NOT EMPTY"
7192 031472 000513                    BR      181             ;SET UP THE ERROR REPORTING ROUTINE.
7193                                     ;GO REPORT THE ERROR.
7194
7195      ; NOW VERIFY THAT NO TX_INTS OCCUR AFTER THE TX ACTION FIFO HAS BEEN EMPTIED,
7196      ; OF TX_ACTIONS, I.E. NO INTERRUPTS OCCUR WITH THE TX ACTION BIT CLEAR.
7197
7198 031474 005001      101:    CLR     R1                ;PASS THE NUMBER OF THE FIRST LINE
7199
7200      ;
7201      ; INITIATE A DMA ON ALL LINES AND WAIT FOR ALL DMA'S TO COMPLETE.
7202
7203 031476 012767 013423 152346      MOV    #5907,ERRNBR    ;SET THE ERROR NUMBER TO 5907.
7204 031504 012702 002650      MOV    #BUFBAS,R2        ;PASS THE START OF THE DMA PATTERN TO TX.
7205 031510 012703 000001      MOV    #1,R3             ;PASS THE LENGTH OF THE DATA PATTERN.
7206 031514 004767 162660      121:    JSR     PC,DODMA        ;TRANSMIT THE DATA PATTERNS.
7207 031520 103102                    BCC     S01          ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
7208 031522 005201                    INC     R1           ;INCREMENT THE LINE NUMBER.
7209 031524 022701 000020      CMP     #16,R1           ;BRANCH TO INITIATE ANOTHER DMA IF,
7210 031530 001371                    BNE     121          ;ALL LINES NOT SERVED.
7211 031532 012704 000144      MOV     #100,R4           ;SET THE DELAY OF 100 MILLI SECS.
7212 031536 004767 162520      JSR     PC,DELAY         ;WAIT FOR THE DMA'S TO COMPLETE.
7213
7214      ;
7215      ; SET UP THE INTERRUPT SERVICE ROUTINE THAT WILL READ THE TX ACTION FIFO
7216      ; UNTIL EMPTY AND CHECK FOR ANY SUBSEQUENT INTERRUPTS WITH NO TX ACTION.
7217      ; ENABLE TX INTERRUPTS.
7218
7218 031542 005067 150524      CLR     TXINTC        ;CLEAR THE TX INT COUNTER.
7219 031546 005067 150522      CLR     TXINTF        ;CLEAR THE TX INT FLAGS.
7220
7221      SETVEC TXVECA,0TXAINT,0PRI06
7222
7222 031552                    MOV     #PRI06,(SP)
7223 031552 012746 000300      MOV     #TXAINT,(SP)
7224 031556 012746 017734      MOV     TXVECA,(SP)
7225 031562 016746 150430      MOV     #3,(SP)
7226 031566 012746 000003      TRAP    C$VEC
7227 031572 104437      ADD     #10,SP
7228 031574 062706 000010      SETPRI  #PRI04        ;ALLOW DEVICE INTERRUPTS.
7229
7229 031600      JSR     PC,TXIE1        ;ENABLE TX INTERRUPTS.
7230
7230 031600 012700 000200      MOV     #PRI04,R0
7231 031604 104441      TRAP    C$SPRI
7232 031606 004767 165444      JSR     PC,TXIE1        ;ENABLE TX INTERRUPTS.
7233
7233      ;
7234      ; WAIT FOR THE INTERRUPTS TO OCCUR
7235
7235      ;
7236      ; SET THE DELAY FOR 5 MILLI SECS.
7237      ; DELAY FOR 5 MS.
7238
7238 031612 012704 000005      MOV     #5,R4
7239 031616 004767 162440      JSR     PC,DELAY
7240
7240      ;
7241      ; DISABLE INTERRUPTS AND CLEAR DOWN THE INTERRUPT SERVICE ROUTINE.
7242
7242 031622      SETPRI  #PRI07        ;DISABLE ALL INTERRUPTS.
7243
7243 031622 012700 000340      MOV     #PRI07,R0
7244 031626 104441      TRAP    C$SPRI
7245
7245 031630 004767 165362      JSR     PC,TXIE0        ;DISABLE OUT TX INTERRUPTS
7246
7246 031634      CLRVEC TXVECA        ;CLEAR THE TX INT VECTOR
7247
7247 031634 016700 150356      MOV     TXVECA,R0
7248 031640 104436      TRAP    C$VEC

```

C15

CZDHVHO DMU 11 FUNC TST PART2 MACRO M1200 15-MAR 84 09:28 PAGE 119-3
HARDWARE TEST TXACTF

SEQ 184

```

7235
7236
7237
7238
7239 031642 005267 152204
7240 031646 005767 150420
7241 031652 001425
7242 031654 005267 152172
7243 031660 005767 150410
7244 031664 100022
7245
7246 031666 012701 007654
7247
7248
7249 031672 012767 011762 152156
7250 031700 000410
7251
7252
7253
7254
7255 031702 010102
7256
7257 031704 012767 013420 152140
7258 031712 012701 000020
7259 031716 012703 007476
7260
7261 031722
7262 031722 104460
7263 031724 000402
7264 031726 004767 164740
7265 031732 005067 150312
7266
7267 031736
7268 031736
7269 031736 104401

;
; VERIFY THAT A TX INTERRUPT OCCURED, AND THAT NO TX INTERRUPT OCCURED WHEN
; THE TX_ACTION FIFO WAS EMPTY.
;
      INC     ERRNBR      ;SET THE ERROR NUMBER TO 5908.
      TST     TXINTC      ;READ THE INTERRUPT COUNT.
      BEQ     501          ;BRANCH TO REPORT ERROR IF NO INTS OCCURED.
      INC     ERRNBR      ;SET THE ERROR NUMBER TO 5909.
      TST     TXINTF      ;READ THE TX INTERRUPT FLAGS.
      BPL     601          ;EXIT WITH SUCCESS IF NO INTERRUPT OCCURED
                          ;WITHOUT A CORRESPONDING TX_ACTION.
                          ;SET UP THE ERROR MESSAGE.
                          ; "TX INT OCCURED AFTER TX FIFO HAD
                          ; BEEN EMPTIED".
      MOV     #ER0503,ERRBLK ;SET THE ERROR ROUTINE IN THE ERROR TABLE.
      BR      181          ;GO AND REPORT THE ERROR.

;
; ERROR REPORTING.
;
141:   MOV     R1,R2      ;PASS THE ACTUAL NUMBER OF TX ACTIONS RECIEVED.
                          ;TO THE ERROR ROUTINE.
      MOV     #5904,,ERRNBR ;SET THE ERROR NUMBER TO 5904.
161:   MOV     #16,,R1    ;PASS THE EXPECTED NUMBER OF TX-ACTIONS.
      MOV     #EM5903,R3  ;SET THE ERROR MESSAGE TO,
                          ; "INCORRECT NUMBER OF TX-ACTIONS FOUND".
181:   ERROR                      ;REPORT THE ERROR.
                          ;
                          ;TRAP    C$ERROR
      BR      601          ;EXIT THE TEST.

501:   JSR     PC,TSABRT    ;REPORT NON RELATED TEST ERROR.
601:   CLR     CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.

      ENDTST

                          L10044:
                          TRAP    C$ETST

```

```

7269 ,SHTTL HARDWARE TEST - TXFIFO -
7270 ;* *****
7271 ;* TXFIFO TEST -
7272 ;* THIS TEST IS USED TO VERIFY THAT THE DUT'S TRANSMIT FIFO'S CAN HOLD
7273 ;* 64 CHARACTERS AND THAT ONLY ONE TX INTERRUPT OCCURS FOR ALL 64
7274 ;* CHARACTERS. THE TEST ALSO EXERCISES THE BYTE SWAPPER BY USING
7275 ;* ALTERNATE WORD AND BYTE WRITES TO THE TX FIFO, AS WELL AS CHECKING THAT
7276 ;* THE FIFOSIZE REGISTER REPORTS THE NUMBER OF CHARACTERS IN THE FIFO
7277 ;* CORRECTLY. ANY BMP CODES FOUND ARE SAVED ON THE QUE TO BE REPORTED
7278 ;* LATER.
7279 ;* THE TEST IS PERFORMED IN INTERNAL LOOPBACK ON ALL ACTIVE LINES.
7280 ;* *****
7281
7282 031740 BGNTST
7283 031740 T16::
7284 031740 SETPRI @PRI05 ;ALLOW LTC INTERRUPTS.
7285 031740 012700 000240 MOV @PRI05,R0
7286 031744 104441 TRAP C$SPRI
7287 031746 000020 TNUM ** TNUM + 1 ;INCREMENT ASSEMBLY TIME TEST COUNTER
7288 031746 012767 000020 150300 MOV @TNUM,TSTNUM ;SET UP THE TEST NUMBER.
7289 031754 012767 177777 150266 MOV @-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
7290 031762 012767 000001 152060 MOV @1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
7291 031770 012767 013561 152054 MOV @6001,ERRNBR ;SET THE ERROR NUMBER TO 6001.
7292 031776 012767 007754 152050 MOV @EM6001,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
7293 032004 005067 150602 CLR ERSNRF ;CLEAR THE ERROR SUMMARY FLAGS.
7294 032010 005067 150236 CLR EXOERR ;CLEAR THE "EXIT ON ERROR" FLAG.
7295 032014 012700 002614 MOV @ERCNTB,R0 ;SET UP THE START ADDRESS FOR THE BLOCK.
7296 032020 004767 162214 JSR PC,CLR16W ;CLEAR THE BLOCK OF 16 WORDS.
7297
7298 ; RESET THE DUT TO A KNOWN STATE, REMOVE ANY STATUS CODES IN THE FIFO.
7299 ; CLEAR THE RX AND TX ENABLE BITS IN THE CSR.
7300 ; THIS SUBROUTINE REPORTS ERROR >>>> 6001 <<<<.
7301
7302 JSR PC,CLNRST ;RESET THE DMU-11 REPORT ANY ERRORS FOUND.
7303 BCS .+6 ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
7304 JMP 621 ;EXIT THE TEST IF FATAL ERROR FOUND.
7305
7306 ; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES AND ENABLE RECEIVERS. SET UP THE
7307 ; LPR'S TO 38.4K BAUD, 8 BITS/CHAR, ODD PARITY, 2 STOP BITS. DISABLE
7308 ; TRANSMITTERS ON ALL LINES.
7309
7310 MOV @MAPLNS,R5 ;INDICATE TO DISABLE ALL LINES.
7311 JSR PC,IXDSBL ;DISABLE TX ON ALL LINES.
7312 MOV ACTLNS,R5 ;INDICATE TO SET UP ACTIVE LINES ONLY.
7313 MOV @204,R0 ;PASS PARAMETER FOR INTERNAL LOOPBACK.
7314 ;ENABLE RECEIVERS.
7315 JSR PC,WTWLCN ;INITIALISE THE LINE CONTROL REGS.
7316 MOV @177670,R0 ;PASS THE LPR CONTENTS.
7317 JSR PC,WTWMPR ;SET THE LPR'S TO 38.4K BAUD.
7318 MOV @10,R4 ;PASS DELAY TIME OF 10 MILLI SECS.
7319 JSR PC,DELAY ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
7320
7321 ; SET UP THE TX INTERRUPT SERVICE ROUTINE AND VECTOR. THE ROUTINE COUNTS
7322 ; THE NUMBER OF INTERRUPTS AND CHECKS FOR ANY INTERRUPTS OCCURING WITH
7323 ; NO TX ACTION.

```

E15

CZDHV80 DHU:11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 120-1
HARDWARE TEST TXFIFO

SEQ 186

```

7323 032102          SETVEC TXVECA,0TXAINT,0PRI06
      032102 012746 000300
      032106 012746 017734
      032112 016746 150100
      032116 012746 000003
      032122 104437
      032124 062706 000010
      032130 004767 165122
7324 032130 004767 165122
7325
7326
7327
7328
7329
7330
7331 032134 016705 150060
7332 032140 005001
7333
7334 032142 000241
7335 032144 006005
7336 032146 103067
7337 032150 110177 150052
7338 032154 010103
7339 032156 000303
7340
7341
7342
7343
7344 032160 012700 000003
7345 032164 010377 150044
7346 032170 105203
7347 032172 110377 150036
7348 032176 062703 000401
7349 032202 005300
7350 032204 001367
7351 032206 005002
7352 032210 117702 150020
7353 032214 122702 000067
7354 032220 001425
7355
7356
7357
7358 032222 010104
7359 032224 012701 000067
7360 032230 010346
7361 032232 012767 013562 151612
7362 032240 012703 010000
7363
7364 032244 012767 012276 151604
7365 032252
      032252 104460
7366 032254 012603
7367 032256 010401
7368 032260 032767 000100 147722
7369 032266 001002
7370 032270 000167 000444
7371
7372

```

```

      MOV TXVECA,0TXAINT,0PRI06
      MOV 0PRI06,(SP)
      MOV 0TXAINT,(SP)
      MOV TXVECA,(SP)
      MOV 03,(SP)
      TRAP C$SVEC
      ADD 010,SP

      JSR PC,TXIE1      ;SET THE TX INT ENABLE BIT IN THE CSR.

      ;
      ; WRITE 64 CHARACTERS TO ALL TXFIFO'S USING ALTERNATE WORD/BYTE WRITES
      ; TO EXERCISE THE BYTE SWAPPER. AFTER THE FIRST 9 CHARACTERS HAVE BEEN
      ; WRITTEN CHECK THAT THE FIFOSIZE REGISTER SHOWS THE CORRECT NUMBER OF
      ; FREE BYTES IN THE FIFO.
      ;
      MOV ACTLNS,R5      ;SET UP THE ACTIVE LINE BIT MAP.
      CLR R1             ;SET UP THE FIRST LINE NUMBER

2$:   CLC                ;CLEAR THE CARRY BIT READY FOR THE ROTATION.
      ROR R5             ;ROTATE THE ACTIVE LINE BIT MAP INTO THE CARRY.
      BCC 10$            ;AVOID TESTING THIS LINE IF ITS INACTIVE.
      MOV R1,0CSRA       ;LOAD THE LINE NUMBER OF THE UNIT INTO THE CSR.
      MOV R1,R3           ;INITIALISE THE DATA PATTERN FOR THIS LINE BY,
      SWAB R3            ;PUTTING THE LINE NUMBER IN THE HIBYTE AND
                        ;CLEARING THE LOBYTE.

      ;
      ; LOAD 9 CHARACTERS INTO THE TXFIFO AND CHECK THE FIFOSIZE REGISTER.
      ;
      MOV 03,R0          ;LOOP COUNT.
4$:   MOV R5,0$DATA      ;MOVE A WORD OF DATA INTO THE FIFO.
      INCB R3            ;INCREMENT THE LOBYTE OF THE DATA PATTERN.
      MOV R3,0$DATA      ;MOVE A BYTE OF DATA INTO THE FIFO.
      ADD 0401,R3        ;INCREMENT THE HIGH AND LOW BYTE OF THE DATA.
      DEC R0             ;DECREMENT THE LOOP COUNT.
      BNE 4$            ;BRANCH IF NOT ALL 9 CHARACTERS WRITTEN.
      CLR R2             ;CLEAR THE UPPER AND LOWER BYTE OF R2.
      MOV 0$SLSA,R2      ;READ THE FIFOSIZE REGISTER.
      CPB 055,R2         ;COMPARE THE EXPECTED SIZE WITH THE ACTUAL.
      BEQ 6$            ;AVOID THE ERROR REPORT IF THE SIZE IS CORRECT.

      ;
      ; REPORT THE ERROR, INCORRECT VALUE. >>>> 6002 <<<<.
      ;
      MOV R1,R4          ;PASS THE LINE NUMBER TO THE ERROR ROUTINE.
      MOV 055,R1         ;PASS THE EXPECTED FIFO SIZE.
      MOV R3,(SP)        ;SAVE THE DATA PATTERN.
      MOV 06002,ERRNBR   ;SET THE ERROR NUMBER TO 6002.
      MOV 0ERM6002,R3    ;PASS THE MESSAGE,
                        ; "INCORRECT VALUE IN FIFOSIZE REGISTER".
      MOV 0ER6001,ERRBLK ;SET THE ERROR REPORTING ROUTINE.
      ERROR

      TRAP C$ERROR

      MOV (SP),R3        ;RESTORE THE PATTERN.
      MOV R4,R1         ;RESTORE THE LINE NUMBER.
      BIT 0BIT06,OPTION  ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
      BNE 6$            ;CONTINUE IF IT HAS.
      JMP 60$           ;EXIT THE TEST IF IT HASN'T.

      ;
      ; CONTINUE FILLING UP THE FIFO UNTIL 64 CHARACTERS HAVE BEEN LOADED.

```

```

7373
7374 032274 012700 000022
7375 032300 010377 147730
7376 032304 105203
7377 032306 110377 147722
7378 032312 062703 000401
7379 032316 005300
7380 032320 001367
7381 032322 110377 147706
7382
7383
7384
7385 032326 005705
7386 032330 001402
7387 032332 005201
7388 032334 000702
7389
7390
7391
7392
7393
7394
7395 032336 016705 147656
7396 032342 005001
7397
7398 032344 000241
7399 032346 006005
7400 032350 103157
7401 032352 110177 147650
7402 032356 010103
7403 032360 000303
7404 032362 012767 013563 151462
7405 032370 004767 162774
7406 032374 103151
7407 032376 005067 147670
7408 032402 005067 147666
7409 032406 052777 100000 147626
7410 032414
      032414 012700 000200
      032420 104441
7411 032422 012704 000144
7412 032426 004767 161630
7413 032432
      032432 012700 000240
      032436 104441
7414 032440 042777 100000 147574
7415
7416
7417
7418
7419 032446 005267 151400
7420 032452 005767 147614
7421 032456 001520
7422 032460 005267 151366
7423 032464 005767 147604
7424 032470 100513
7425 032472 022767 000001 147572

```

```

;
68:  MOV  #18.,R0      ;SET UP THE OUTER LOOP.
88:  MOV  R3,&FDATA     ;MOVE A WORD OF DATA INTO THE FIFO.
      INCB  R3          ;INCREMENT THE LOBYTE OF THE DATA PATTERN.
      MOVB  R3,&FDATA   ;MOVE A BYTE OF DATA INTO THE FIFO.
      ADD  #401,R3     ;INCREMENT THE HIGH AND LOW BYTE OF THE DATA.
      DEC  R0          ;DECREMENT THE LOOP COUNT.
      BNE  #1          ;BRANCH IF NOT ALL 54 CHARACTERS WRITTEN.
      MOVB  R3,&FDATA   ;LOAD THE LAST CHARACTER.
;
; LOOP UNTIL THE TXFIFO'S ON ALL ACTIVE LINES HAVE BEEN FILLED.
;
108:  TST  R5           ;HAVE ALL FIFOS BEEN FILLED.
      BEQ  128         ;YES, THEN GO AND TRANSMIT THE CHARACTERS.
      INC  R1          ;OTHERWISE SELECT THE NEXT LINE.
      BR   28          ;AND BRANCH TO TEST IF ITS ACTIVE.
;
; ENABLE TRANSMISSION ON EACH LINE IN TURN AND WAIT FOR 100 MS TO ALLOW
; ANY INTERRUPTS TO OCCUR BEFORE DISABLING TRANSMISSION. VERIFY THAT ONLY
; INTERRUPT OCCURED AND CHECK THAT THE CHARACTERS WERE TRANSMITTED UNCORRUPTED
; AND ON THE CORRECT LINE.
;
128:  MOV  ACTLNS,R5    ;SET UP THE ACTIVE LINE BIT MAP.
      CLR  R1          ;SET UP THE FIRST LINE NUMBER.
;
148:  CLC              ;CLEAR THE CARRY BIT READY FOR THE ROTATION.
      ROR  R5          ;ROTATE THE ACTIVE LINE BIT MAP INTO THE CARRY.
      BCC  248         ;AVOID TESTING THIS LINE IF ITS INACTIVE.
      MOVB  R1,&CSRA    ;LOAD THE LINE NUMBER OF THE OUT INTO THE CSR.
      MOV  R1,R3        ;INITIALISE THE "EXPECTED" DATA PATTERN FOR
      SWAB  R3          ;THIS LINE.
      MOV  #6003.,ERRNBR ;SET THE ERROR NUMBER TO 6003.
      JSR  PC,PUFIFO    ;PURGE THE RXFIFO.
      BCC  508         ;REPORT THE ERROR IF THE FIFO WOULD NOT PURGE.
      CLR  TXINTC       ;CLEAR THE INTERRUPT COUNT.
      CLR  TXINTF       ;CLEAR THE INTERRUPT FLAGS.
      BIS  #BIT15,&TXAD2A ;ENABLE TRANSMISSION ON THIS LINE.
      SETPRI #PRI04     ;ALLOW DEVICE INTERRUPTS.
;
;
      MOV  #PRI04,R0    ;
      TRAP C$SPRI
;
      MOV  #100.,R4     ;SET THE DELAY.
      JSR  PC,DELAY     ;DELAY FOR 100 MILLI SECS.
      SETPRI #PRI05
;
;
      MOV  #PRI05,R0    ;
      TRAP C$SPRI
;
      BIC  #BIT15,&TXAD2A ;DISABLE TRANSMISSION ON THIS LINE.
;
; VERIFY THAT ONLY ONE INTERRUPT OCCURED AND THAT IT WAS ACCOMPANIED BY
; A CORRESPONDING TX ACTION.
;
;
      INC  ERRNBR       ;SET THE ERROR NUMBER TO 6004.
      TST  TXINTC       ;HAS AN INTERRUPT OCCURED ?
      BEQ  508         ;REPORT THE ERROR IF NO INTERRUPT OCCURED.
      INC  ERRNBR       ;SET THE ERROR NUMBER TO 6005.
      TST  TXINTF       ;HAS AN INTERRUPT OCCURED WITHOUT A TX ACTION ?
      BMI  508         ;REPORT THE ERROR IF IT HAS.
      CMP  #1,TXINTC    ;DID ONLY ONE INT OCCUR ?

```

```

7426 032500 001424          BEQ      16$          ;AVOID THE ERROR IF ONLY ONE INTERRUPT.
7427
7428          ;* REPORT THE ERROR, MORE THAN ONE INTERRUPT. >>>> 6006 <<<<.
7429          ;-
7430 032502 016702 147564      MOV      TXINTC,R2      ;PASS THE ACTUAL NUMBER OF INTERRUPTS.
7431 032506 010104              MOV      R1,R4          ;PASS THE LINE NUMBER.
7432 032510 012701 000001      MOV      #1,R1          ;PASS THE EXPECTED NUMBER OF INTS.
7433 032514 010346              MOV      R3,-(SP)        ;SAVE THE DATA PATTERN.
7434 032516 012703 010040      MOV      @EM6003,R3      ;PASS THE MESSAGE.
7435
7436              ; "MORE THAN ONE TX-INT OCCURED FROM A FULL
7437 032522 005267 151324      INC      ERRNBR          ;SET THE ERROR NUMBER TO 6006.
7438 032526 012767 012276 151322      MOV      @ER6001,ERRBLK ;SET UP THE ERROR BLOCK.
7439 032534
7440 032536 012603              MOV      (SP)+,R3        ;RESTORE THE DATA PATTERN.
7441 032540 010401              MOV      R4,R1          ;RESTORE THE LINE NUMBER.
7442 032542 032767 000100 147440      BIT      @BIT06,OPTION ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
7443 032550 001473              BEQ      60$          ;EXIT THE TEST IF IT HAS.
7444
7445
7446          ;* READ THE CHARACTERS FROM THE RXFIFO AND VERIFY THEY ARE CORRECT AND WERE
7447          ; RECIEVED ON THE CORRECT LINE.
7448          ; THIS SUBROUTINE REPORTS ERRORS, >>>> 6007 THRU 6008 <<<<.
7449          ;-
7450
7451 032552 012767 013567 151272 16$:      MOV      @6007,ERRNBR ;SET UP THE ERROR NUMBER TO 6007.
7452 032560 005000              CLR      R0              ;INITIALISE THE NUMBER OF CHARS READ COUNT.
7453 032562 012704 000025      MOV      @21,R4          ;SET UP THE OUTER LOOP COUNT.
7454 032566 012702 000003 18$:      MOV      #3,R2          ;SET UP THE INNER LOOP COUNT.
7455 032572 004767 163140 20$:      JSR      PC,REPDER ;READ A CHARACTER FROM THE RXFIFO, VERIFY THAT
7456
7457              ;IT IS CORRECT AND CAME FROM THE OUT. REPORT
7458 032576 103022              BCC      22$          ;ANY ERRORS. >>>> 6007 THRU 6008 <<<<.
7459 032600 005767 147446      TST      EXOERR          ;BRANCH TO REPORT THE ERROR IF THE FIFO EMPTY.
7460 032604 001055              BNE      60$          ;TEST THE "EXIT ON ERROR" FLAG.
7461
7462 032606 005200              INC      R0              ;EXIT THE TEST IF SET, I.E. AN ERROR OCCURED.
7463 032610 105203              INCB     R3              ;AND NO EXTENDED ERROR REPORTING WAS REQUESTED.
7464 032612 000303              SWAB     R3              ;INCREMENT THE READ CHAR COUNT.
7465 032614 005302              DEC      R2              ;INCREMENT THE LOBYTE OF THE DATA PATTERN.
7466 032616 001365              BNE      20$          ;SWAP BYTES TO PLACE EXPECTED CHAR IN LOBYTE.
7467
7468 032620 000303              SWAB     R3              ;DECREMENT THE INNER LOOP COUNTER.
7469 032622 005304              DEC      R4              ;BRANCH TO READ ANOTHER CHAR IF 3 CHARS HAVE
7470 032624 001360              BNE      18$          ;NOT BEEN READ.
7471
7472 032626 004767 163104      JSR      PC,REPDER ;RESTORE THE DATA BYTES TO THE CORRECT POSITION
7473 032632 103004              BCC      22$          ;DECREMENT THE OUTER LOOP COUNTER.
7474 032634 005767 147412      TST      EXOERR          ;BRANCH TO READ ANOTHER 3 CHARACTERS IF NOT
7475 032640 001037              BNE      60$          ;ALL 63 HAVE BEEN READ.
7476 032642 000422              BR       24$          ;READ AND CHECK THE LAST CHARACTER.
7477
7478          ;* REPORT THE ERROR, NOT ALL CHARACTERS TRANSMITTED. >>>> 6009 <<<<.
7479          ;-
7480 032644 012767 013571 151200 22$:      MOV      @6009,ERRNBR ;SET THE ERROR NUMBER TO 6009.
7481 032652 012703 010243      MOV      @EM6006,R3 ;PASS THE MESSAGE,

```

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</
---------	---------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	----------

```

7511 .SBTTL  HARDWARE TEST - BREAKB -
7512 ;*****
7513 ;* - BREAK GENERATION TEST -
7514 ;* THIS TEST VERIFIES THAT ALL SERIAL TRANSMIT LINES CAN GENERATE A BREAK
7515 ;* BY SETTING THE BRK BIT IN THE ASSOCIATED LNCRTL REGISTER.
7516 ;* USE OF THE INTERNAL LOOPBACK FEATURE OF THE DUARTS IS MADE TO MINIMISE
7517 ;* ANY EXTERNAL EFFECTS CAUSED ON THE SERIAL LINES BY THIS TEST
7518 ;* FRAMING ERROR DETECTION IS USED TO INDICATE THE PRESENCE OF A BREAK,
7519 ;* BY SETTING THE APPROPRIATE BIT IN THE RBUF REGISTER.
7520 ;*****
7521
7522 032766 BGNTEST
7523
7524 032766 012767 177777 147254 MOV #1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
7525 000021 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
7526 032774 012767 000021 147252 MOV #TNUM,ISTNUM ;SET UP THE TEST NUMBER. (64)
7527 033002 012767 000001 151040 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
7528 033010 012767 014401 151034 MOV #6401,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
7529 033016 012767 010307 151030 MOV #EM6401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTABL.
7530
7531 ;* RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
7532 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
7533 ;* THIS SUBROUTINE REPORTS ERROR >>>> 6401 <<<<.
7534
7535 033024 004767 161166 JSR PC,CLNRST ;RESET THE DHU-11. REPORT ANY ERRORS FOUND.
7536 033030 103165 BCC 601 ;EXIT TEST IF FATAL ERROR FOUND.
7537
7538 ;* SET UP DEVICE UNDER TEST (DUT) TO:
7539 ;* DISABLE TRANSMISSION AND RECEPTION INTERRUPTS.
7540 ;* DELAY FOR 10 MILLI-SECONDS TO ALLOW TIME TO CLEAR ANY BREAKS.
7541
7542 033032 012705 177777 MOV #MAPLNS,R5 ;PASS ACTIVE LINE BIT MAP.
7543 033036 012700 000200 MOV #200,R0 ;PASS INTERNAL LOOPBACK MODE.
7544 033042 004767 164504 JSR PC,WTWLCN ;SELECT INTERNAL LOOPBACK,DISABLE DMA.
7545 033046 012704 000012 MOV #10,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
7546 033052 004767 161204 JSR PC,DELAY ;DELAY TO ALLOW ANY BREAKS TO BE CLEARED.
7547
7548 ;* SET UP TRANSMISSION AND RECEPTION PARAMETERS FOR ALL LINES.
7549 ;* 9600 BAUD,8 CHAR,1 STOPBIT,NO PARITY.
7550
7551 033056 012700 156430 MOV #156430,R0 ;SET UP BAUD RATE,ETC.
7552 033062 004767 164514 JSR PC,WTWLPR ;SET COMMUNICATION PARAMETERS ON ALL LINES.
7553
7554 ;* ENABLE TRANSMITTERS ON ALL ACTIVE LINES.
7555
7556 033066 016705 147126 MOV ACTLNS,R5 ;PASS ACTIVE LINE BIT MAP.
7557 033072 004767 164024 JSR PC,TXENBL ;ENABLE TRANSMISSIONS ON ALL LINES.
7558
7559 ;* PURGE THE FIFO OF ANY UNWANTED CHARACTERS.
7560 ;* THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>> 6402 THRU 6404 <<<<.
7561
7562 033076 005267 150750 INC ERRNBR ;SET ERROR NUMBER TO 6402.
7563 033102 004767 162344 JSR PC,PUFIFR ;PURGE FIFO.
7564 033106 103136 BCC 601 ;ABORT TEST IF FIFO WILL NOT CLEAR.
7565
7566 ;* VERIFY BREAK GENERATION ON INDIVIDUAL LINES.

```

```

7567      ; CLEAR BREAKS ON ALL LINES.
7568      ; DELAY FOR 10 MILLI-SECONDS TO ALLOW TIME FOR ANY BREAKS TO BE CLEARED.
7569      ; SELECT LINE, SET BREAK BIT IN LNCTRL REGISTER.
7570      ; TEST FOR A CHARACTER IN THE FIFO WITH FRAME ERROR.
7571      ;
7572 033110 005002      2$: CLR R2 ;CLEAR LINE COUNTER.
7573 033112 012703 000001      MOV #1,R3 ;SET UP ACTIVE LINE BIT MASK.
7574 033116 030367 147076      4$: BIT R3,ACTLNS ;CHECK IF THIS LINE IS ACTIVE.
7575 033122 001440      BEQ #1 ;GO SELECT NEXT LINE IF THIS ONE IS INACTIVE.
7576 033124 012700 000200      MOV #200,R0 ;SET UP PARAMETER TO CLEAR BREAK BITS.
7577 033130 004767 164416      JSR PC,WTLNC ;CLEAR BREAK BIT, RESELECT INTERNAL LOOPBACK.
7578 033134 012704 000012      MOV #10,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
7579 033140 004767 161116      JSR PC,DELAY ;DELAY TO ALLOW BREAKS TO BE CLEARED.
7580      ;
7581      ; SET BREAK BIT ON SELECTED LINE.
7582      ; SET UP PARAMETERS TO TEST FOR THE FRAME ERROR BIT SET IN RBUF.
7583      ; TIME-OUT = 5 MILLI SECONDS.
7584      ; CALL ROUTINE TO CHECK FOR CONDITION FOUND.
7585      ;
7586 033144 010305      6$: MOV R3,R5 ;COPY ACTIVE LINE BIT MASK.
7587 033146 012700 000214      MOV #214,R0 ;SET BREAK, RESELECT LOOPBACK, ENABLE RECEPTION.
7588 033152 004767 164374      JSR PC,WTLNC ;SET BREAK ON SELECTED LINE.
7589      ;
7590      ; DELAY FOR 5 MS TO ALLOW TIME FOR BREAK TO BE GENERATED AND RECEIVED.
7591      ; VERIFY RECEPTION OF A CHARACTER WITH FRAME ERROR BIT SET.
7592      ;
7593 033156 012704 000005      MOV #5,R4 ;SET DELAY VALUE TO 5 MILLI SECS.
7594 033162 004767 161074      JSR PC,DELAY ;ALLOW TIME FOR CHARACTER RECEPTION.
7595 033166 017700 147036      MOV #RBUF,R0 ;GET CHARACTER FROM RBUF REGISTER.
7596 033172 032700 020000      BIT #BIT13,R0 ;CHECK FOR FRAME ERROR BIT.
7597 033176 001012      BNE #1 ;SKIP ERROR REPORT IF SET.
7598 033200 012701 010344      MOV #EM6402,R1 ;SELECT MESSAGE TO BE PRINTED.
7599      ;REPORT ERROR "BREAK NOT RECEIVED ON LINE #NN"
7600 033204      ERDF 6405,EM6401,ER6401 ; >>>>> ERROR #6405 <<<<<.
7601      ;
7602      ;
7603      ;
7604      ;
7605 033214 032767 000100 146766      BIT #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
7606 033222 001470      BEQ #60 ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7607      ; DURING THE SOFTWARE QUESTIONS.
7608      ;
7609 033224 006303      8$: ASL R3 ;SHIFT BIT MASK FOR NEXT LINE.
7610 033226 005202      INC R2 ;NEXT LINE
7611 033230 020227 000020      CMP R2,#NUMLNS ;CHECK FOR MAX LINE COUNT.
7612 033234 001330      BNE #4 ;IF <>, LOOP TO CHECK NEXT LINE
7613      ;
7614      ; VERIFY BREAK GENERATION ON ALL LINES SIMULTANEOUSLY.
7615      ; CLEAR BREAKS ON ALL LINES.
7616      ; DELAY FOR 10 MILLI-SECONDS TO ALLOW TIME FOR ANY BREAKS TO BE CLEARED.
7617      ; PURGE THE FIFO.
7618      ; SET BREAK BIT IN LNCTRL REGISTERS ON ALL ACTIVE LINES.
7619      ; TEST FOR CHARACTERS IN THE FIFO WITH FRAME ERROR.

```

```

TRAP C$ERDF
WORD 640
WORD EM6401
WORD ER6401

```

```

7620
7621 033236 012705 177777      1-      MOV      #MAPLNS,R5      ;SET UP LINE TO CLEAR BREAKS ON.
7622 033242 012700 000200      MOV      #200,R0      ;SET UP PARAMETER TO CLEAR BREAK BITS.
7623 033246 004767 164300      JSR      PC,WTWLNLC      ;CLEAR BREAK BIT,RESELECT INTERNAL LOOPBACK.
7624 033252 012704 000012      MOV      #10.,R4      ;PASS DELAY TIME OF 10 MILLI SECONDS.
7625 033256 004767 161000      JSR      PC,DELAY      ;DELAY TO ALLOW BREAKS TO BE CLEARED.
7626
7627      ;+
7628      ; PURGE THE FIFO OF UNWANTED CHARACTERS.
7629 033262 004767 162102      JSR      PC,PUFIFO      ;PURGE FIFO.
7630 033266 103044      BCC      50$      ;GO REPORT ERROR IF FAILED TO CLEAN OUT FIFO.
7631
7632      ;+
7633      ; SET UP PARAMETERS FOR SETTING THE BREAK BIT ON ALL ACTIVE LINES.
7634      ; THEN CALL ROUTINE TO DO IT.
7635 033270 016705 146724      10$:      MOV      ACTLNS,R5      ;SET UP ACTIVE LINE BIT MASK.
7636 033274 012700 000214      MOV      #214,R0      ;SET BREAK,RESELECT LOOPBACK,ENABLE RECEPTION.
7637 033300 004767 164246      JSR      PC,WTWLNLC      ;SET BREAK ON SELECTED LINES.
7638
7639      ;+
7640      ; DELAY FOR 10 MILLI SECONDS,TO ALLOW TIME FOR RECEPTION.
7641      ; TEST FOR CHARACTERS IN FIFO WITH FRAME ERROR BIT SET.
7642 033304 012704 000012      1-      MOV      #10.,R4      ;SET DELAY VALUE TO 10 MILLI SECS.
7643 033310 004767 160746      JSR      PC,DELAY      ;ALLOW TIME FOR CHARACTER RECEPTION.
7644 033314 010502      MOV      R5,R2      ;COPY ACTIVE LINE BIT MAP.
7645 033316 004767 161354      JSR      PC,MAPCNT      ;COUNT THE NUMBER OF LINES AVAILABLE.
7646 033322 017701 146702      12$:      MOV      #RBUFA,R1      ;GET CHARACTER FROM RBUF REGISTER.
7647 033326 100011      BPL      14$      ;BRANCH IF DATA_VALID NOT SET.
7648 033330 032701 020000      BIT      #BIT13,R1      ;CHECK FOR FRAME ERROR BIT.
7649 033334 001406      BEQ      14$      ;DO NOT CLR FLG FOR THIS LINE IF FRAME BIT CLR.
7650 033336 000301      SWAB      R1      ;GET LINE NUMBER IN LOW BYTE.
7651 033340 042701 177400      BIC      #177400,R1      ;CLEAR EVERYTHING BUT THE LINE NUMBER.
7652 033344 004767 161300      JSR      PC,LINBIT      ;CALC BIT MASK FROM LINE NUMBER.
7653 033350 040005      BIC      R0,R5      ;CLEAR LINE FLAG.
7654 033352 005302      14$:      DEC      R2      ;DECREMENT THE LINE NUMBER COUNTER.
7655 033354 001362      BNE      12$      ;LOOP TO GET THE NEXT CHARACTER.
7656 033356 005705      TST      R5      ;CHECK IF ANY BREAKS NOT RECEIVED.
7657 033360 001411      BEQ      60$      ;EXIT TEST IF ALL CLEAR.
7658 033362 012701 010344      MOV      #EM6402,R1      ;SELECT MESSAGE TO BE PRINTED.
7659      ;REPORT ERROR"BREAK NOT RECEIVED ON LINE #NN".
7660 033366      ERRDF      6406,EM6401,ER6401;      >>>>> ERROR #6407 <<<<<.
7661 033370      TRAP      C$ERDF
7662 033372      .WORD      6406
7663 033374      .WORD      EM6401
7664 033376      .WORD      ER6401
7665 033400      BR      60$      ;EXIT THE TEST.
7666 033404      50$:      JSR      PC,TSABRT      ;ABORT THE TEST.
7667 033410      60$:      CLR      CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7668 033410      ENDTST
7669 033410      L10046:      TRAP      C$ETST
7670 033410      104401

```

```

7667 .SBTTL  HARDWARE TEST - NORERR -
7668 ;*****
7669 ; - NO OVERRUN ERROR TEST -
7670 ;
7671 ; THIS TEST VERIFIES THAT THE DUT WILL NOT REPORT DATA OVERRUN
7672 ; ERRORS WHEN THEY DO NOT OCCUR.
7673 ; THIS TEST PUTS 256 CHARACTERS IN THE DUT FIFO PLUS 4 IN EACH ACTIVE
7674 ; UART AND VERIFIES THAT NO OVERRUN ERRORS ARE REPORTED.
7675 ; ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
7676 ; HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
7677 ; REPORTED LATER.
7678 ;
7679 ;-----*****
7680
7681 033412          BGNTST
7682 033412          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.          T18::
7683 033412 012700 000240          MOV  #PRI05,R0
7684 033416 104441          TRAP  C$SPRI
7685 033412 000022          TNUM = TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
7686 033420 012767 000022 146626  MOV  #TNUM,TSTNUM          ;SET UP THE TEST NUMBER. (66)
7687 033426 012767 177777 146614  MOV  #-1,CTRLCF          ;INDICATE THAT WE ARE IN A TEST.
7688 033434 012767 000001 150406  MOV  #1,ERRTYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
7689 033442 012767 014711 150402  MOV  #6601,ERRNBR          ;SET ERROR NUMBER TO 6601.
7690 033450 012767 010405 150376  MOV  #EM6601,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
7691 ;
7692 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
7693 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
7694 ; THIS SUBROUTINE REPORTS ERROR >>>> 6601 <<<<.
7695 ;
7696 033456 004767 160534          JSR  PC,CLNRST          ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
7697 033462 103402          BCS  .+6          ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
7698 033464 000167 000432          JMP  60$          ;EXIT THE TEST, FATAL ERROR WAS FOUND.
7699 ;
7700 ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
7701 ; INITIALIZE THE 256 BYTE DATA PATTERN.
7702 ;
7703 033470 004767 160774          JSR  PC,FINACT          ;FIND AN ACTIVE LINE.
7704 033474 103402          BCS  .+6          ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
7705 033476 000167 000420          JMP  60$          ;EXIT THE TEST, FATAL ERROR WAS FOUND.
7706 033502 004767 161042          JSR  PC,INDATP          ;INITIALISE DATA PATTERN.
7707 ;
7708 ; TRANSMIT A 265 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
7709 ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
7710 ;
7711 ; SET INTERNAL LOOPBACK ON THE SELECTED LINE.
7712 ; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
7713 ;
7714 033506 005267 150340          INC  ERRNBR          ;SET THE ERROR REPORT NUMBER TO 6602.
7715 033512 012700 000204          MOV  #204,R0          ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
7716 033516 004767 164030          JSR  PC,WTWLC          ;INITIALISE THE LINE CONTROL REGISTER.
7717 033522 012700 177670          MOV  #177670,R0          ;PASS THE LPR CONTENTS.
7718 033526 004767 164050          JSR  PC,WTWLPR          ;SET THE LPR CONTENTS TO 38.4K BAUD.
7719 033532 012704 000012          MOV  #10,R4          ;PASS DELAY TIME OF 10 MILLI SECONDS.
7720 033536 004767 160520          JSR  PC,DELAY          ;WAIT FOR LNCRTL AND LPR REGS TO BE UPDATED.
7721 033542 012702 002650          MOV  #BUF8AS,R2          ;PASS THE START OF THE DATA PATTERN TO TX.

```

```

7721 033546 012703 000400      MOV    #BUFMID-BUFBAS,R3 ;PASS THE LENGTH OF THE DATA PATTERN,
7722 033552 004767 160622      JSR     PC,DODMA      ;TRANSMIT THE DATA PATTERN.
7723 033556 103157              BCC     50$      ;EXIT IF ERROR FOUND DURING DMA TX.
7724
7725      ;+
7726      ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
7727      ; THE FIFO.
7728      ;+
7728 033560 005267 150266      INC     ERRNBR      ;SET ERROR NUMBER TO 6603.
7729 033564 012701 170536      MOV     #170536,R1      ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
7730 033570 016702 146432      MOV     CSRA,R2      ;PASS THE ADDRESS OF THE CSR.
7731 033574 004767 163636      JSR     PC,WAIBIS      ;WAIT FOR DMA TO COMPLETE, TX ACTION SET.
7732 033600 103146              BCC     50$      ;ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
7733 033602 012704 000005      MOV     #5,R4      ;PASS DELAY OF 5 MILLI SECS.
7734 033606 004767 160450      JSR     PC,DELAY      ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
7735
7736
7737      ;+
7738      ; TRANSMIT 4 CHARACTERS ON EACH ACTIVE LINE.
7739      ;+
7739 033612 016705 146402      MOV     ACTLNS,R5      ;ALTER PARAMETERS FOR ALL ACTIVE LINES.
7740 033616 012700 000204      MOV     #204,R0      ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
7741 033622 004767 163724      JSR     PC,WTWLNC      ;INITIALISE THE LINE CONTROL REGISTER.
7742 033626 012700 177670      MOV     #177670,R0      ;PASS THE LPR CONTENTS.
7743 033632 004767 163744      JSR     PC,WTWLPR      ;SET THE LPR CONTENTS TO 38.4K BAUD.
7744 033636 012704 000012      MOV     #10,R4      ;PASS DELAY TIME OF 10 MILLI SECONDS.
7745 033642 004767 160414      JSR     PC,DELAY      ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
7746
7747 033646 012702 002650      MOV     #BUFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
7748 033652 012703 000004      MOV     #4,R3      ;PASS THE LENGTH OF THE DATA PATTERN.
7749 033656 005001              CLR     R1      ;CLEAR THE LINE COUNTER.
7750 033660 005267 150166      INC     ERRNBR      ;SET ERROR NUMBER TO 6604.
7751 033664 010100              2$:    MOV     R1,R0
7752 033666 006300              ASL     R0      ;CALCULATE THE LINE OFFSET FROM THE LINE #.
7753 033670 036067 002336 146322  BIT     BITTBL(R0),ACTLNS ;TEST FOR THIS LINE BEING ACTIVE.
7754 033676 001403              BEQ     4$      ;SKIP THE TX ON THIS LINE IF IT IS NOT ACTIVE.
7755 033700 004767 160474      JSR     PC,DODMA      ;TRANSMIT THE 5 CHAR DATA PATTERN.
7756 033704 103104              BCC     50$      ;ABORT IF ERROR FOUND DURING DMA TX.
7757 033706 005201              4$:    INC     R1      ;INCREMENT THE LINE COUNTER.
7758 033710 020127 000020      CMP     R1,#NUMLNS      ;TEST FOR ALL POSSIBLE LINES HANDLED
7759 033714 002763              BLT     2$      ;LOOP IF NOT ALL LINES HANDLED.
7760
7761 033716 005267 150130      INC     ERRNBR      ;SET ERROR NUMBER TO 6605.
7762 033722 012701 170040      MOV     #170040,R1      ;PASS TIME-OUT VALUE OF 32 MILLI SECS.
7763 033726 016702 146274      MOV     CSRA,R2      ;PASS THE ADDRESS OF THE CSR.
7764 033732 004767 163500      JSR     PC,WAIBIS      ;WAIT FOR A DMA TO COMPLETE, TX ACTION SET.
7765 033736 103067              BCC     50$      ;ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
7766 033740 012704 000005      MOV     #5,R4      ;PASS DELAY OF 5 MILLI SECS.
7767 033744 004767 160312      JSR     PC,DELAY      ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
7768
7769      ;+
7770      ; READ THE FIFO CHECKING FOR OVERRUN ERRORS. REPORT ERRORS IF FOUND.
7771      ; ABORT THE TEST IF A BMP CODE WAS FOUND IN THE FIFO.
7772      ;+
7772 033750 016702 146244      MOV     ACTLNS,R2
7773 033754 004767 160716      JSR     PC,MAPCNT      ;GET THE NUMBER OF ACTIVE LINES.
7774 033760 006302              ASL     R2
7775 033762 006302              ASL     R2      ;MULTIPLY NUMBER OF ACTIVE LINES BY 4.
7776 033764 012705 000400      MOV     #256,R5
7777 033770 060205              ADD     R2,R5      ;CALCULATE NUMBER OF CHARACTERS TO RX.

```

```

7778 033772 005004          CLR    R4          ;CLEAR THE CHARACTER COUNTER.
7779 033774 012767 014716 150050 6$:    MOV    $6606,,ERRNBR ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
7780 034002 017702 146222          MOV    $RBUFA,R2 ;READ A CHARACTER FROM THE FIFO.
7781 034006 100036          BPL     10$          ;EXIT THE READ LOOP IF THE FIFO IS EMPTY.
7782
7783          ;+
7784          ; CHECK IF THE READ CHARACTER IS A BMP CODE.
7785          ; IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
7786          ; ABORT THE TEST.
7787 034010 004767 157762          JSR     PC,CHKBMP ;CHECK IF CHARACTER IS A BMP CODE.
7788 034014 103002          BCC     8$          ;BRANCH IF NOT A BMP CODE.
7789 034016 104460          ERROR          ;
7790 034020 000440          BR      60$          ;EXIT THIS TEST.
7791
7792 034022 005267 150024          8$:    INC     ERRNBR ;SET ERROR NUMBER TO 6607.
7793 034026 005204          INC     R4          ;COUNT THIS CHARACTER.
7794 034030 020405          CMP     R4,R5        ;COMPARE # OF CHARS WITH MAX # OF CHARS.
7795 034032 003031          BGT     50$          ;ABORT TEST IF TOO MANY VALID CHARS READ.
7796 034034 032702 040000          BIT     $BIT14,R2 ;TEST THE OVERRUN BIT OF THE READ CHAR.
7797 034040 001755          BEQ     6$          ;LOOP TO READ THE NEXT CHAR IF NO ERROR.
7798 034042 005267 150004          INC     ERRNBR ;SET ERROR NUMBER TO 6608.
7799 034046 012767 012506 150002          MOV    $ER7801,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
7800 034054 012701 010442          MOV    $EM6602,R1 ;PASS THE MESSAGE TO BE REPORTED.
7801 034060 010203          MOV    R2,R3
7802 034062 000303          SWAB    R3
7803 034064 042703 177760          BIC     $177760,R3 ;GET FAILING LINE NUMBER.
7804          ;REPORT "OVERRUN ERROR REPORTED WHEN NONE FORCED, ON LINE NN ..."
7805 034070          ERROR          ;
7806 034070 104460          ;
7807          ;
7808          ;+
7809          ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
7810 034072 032767 000100 146110          BIT     $BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
7811 034100 001410          BEQ     60$          ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
7812          ; DURING THE SOFTWARE QUESTIONS.
7813
7814 034102 000734          BR      6$          ;LOOP TO READ THE NEXT CHAR.
7815
7816 034104 012767 014721 147740 10$:    MOV    $6609,,ERRNBR ;SET ERROR NUMBER TO 6609.
7817 034112 020405          CMP     R4,R5        ;COMPARE NUMBER OF CHARS READ WITH EXPECTED.
7818 034114 001402          BEQ     60$          ;EXIT TEST WITHOUT ABORT IF CORRECT # OF CHARS.
7819
7820 034116 004767 162550          50$:    JSR     PC,TSABRT ;ABORT THE TEST, NON-RELATED TEST ERROR FOUND.
7821 034122 005067 146122          60$:    CLR     CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7822 034126          ENDTST
7823 034126          L10047:
7824 034126 104401          TRAP     C$ETST

```

```

7826 .SBTTL HARDWARE TEST - ORERR
7827 .....
7828 - OVERRUN ERROR TEST -
7829
7830 THIS TEST VERIFIES THAT THE DUT WILL REPORT DATA OVERRUN ERRORS WHEN
7831 THEY OCCUR.
7832 THIS TEST PUTS 256 CHARACTERS IN THE DUT FIFO PLUS 5 IN EACH ACTIVE
7833 UART AND VERIFIES THAT OVERRUN ERRORS ARE REPORTED ON ALL ACTIVE LINES.
7834 ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
7835 HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
7836 REPORTED LATER.
7837 .....
7838 BGN TST
7839
7840          034130          012700 000240          T19::
7841          034130          104441          MOV      @PRI05,R0
7842          034130          000023          TRAP     C1SPRI
7843          034134          000023          INUM == INUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
7844          034136          012767 000023 146110      MOV      @INUM,TSTNUM      ;SET UP THE TEST NUMBER. (67)
7845          034144          012767 177777 146076      MOV      @-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
7846          034152          012767 000001 147670      MOV      @1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
7847          034160          012767 015055 147664      MOV      @6701,ERRNBR      ;SET ERROR NUMBER TO 6701.
7848          034166          012767 010514 147660      MOV      @EM6701,ERRMSG      ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
7849
7850 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
7851 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
7852 ; THIS SUBROUTINE REPORTS ERROR >>>> 6701 <<<<.
7853
7854          034174          004767 160016      JSR      PC,CLNRST      ;RESET THE DMU 11, REPORT ANY ERRORS FOUND.
7855          034200          103402          BCS      .+6      ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
7856          034202          000167 000660      JMP      601      ;EXIT THE TEST, FATAL ERROR WAS FOUND.
7857
7858 ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
7859 ; INITIALIZE THE 256 BYTE DATA PATTERN.
7860
7861          034206          004767 160256      JSR      PC,FINACT      ;FIND AN ACTIVE LINE.
7862          034212          103402          BCS      .+6      ;IF ACTIVE LINE IS FOUND, DON'T ABORT TEST.
7863          034214          000167 000646      JMP      601      ;ABORT THE TEST, NO ACTIVE LINES WERE FOUND.
7864          034220          004767 160324      JSR      PC,INDATP      ;INITIALISE DATA PATTERN.
7865
7866 ; TRANSMIT A 255 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
7867 ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
7868
7869 ; SET INTERNAL LOOPBACK ON THE SELECTED LINE.
7870 ; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
7871
7872          034224          005267 147622      INC      ERRNBR      ;SET ERROR NUMBER TO 6702.
7873          034230          012700 000204      MOV      @204,R0      ;PASS PARAMETER FOR INTERNAL LOOPBACK,ENABLE RX.
7874          034234          004767 163312      JSR      PC,WTLNLC      ;INITIALISE THE LINE CONTROL REGISTER.
7875          034240          012700 177670      MOV      @177670,R0      ;PASS THE LPR CONTENTS.
7876          034244          004767 163332      JSR      PC,WTLNLR      ;SET THE LPR CONTENTS TO 38.4K BAUD.
7877          034250          012704 000012      MOV      @10,R4      ;PASS DELAY TIME OF 10 MILLI SECONDS.
7878          034254          004767 160002      JSR      PC,DELAY      ;WAIT FOR INCTL AND LPR REGS TO BE UPDATED.
7879          034260          012702 002650      MOV      @BUFBA1,R2      ;PASS THE START OF THE DATA PATTERN TO TX.

```

C16

C:\DMYHO DMU 11 FUNC TST PART2 MACRO M1200 15-MAR-84 09:28 PAGE 123-1

SEQ 197

7878	034264	012703	000400		MOV	#BUF MID BUF BAS, R3	; PASS THE LENGTH OF THE DATA PATTERN.
7879	034270	004767	160104		JSR	PC, DODMA	; TRANSMIT THE DATA PATTERN.
7880	034274	103402			BCS	.+6	; IF NO ERROR FOUND DURING DMA TX, DON'T ABORT.
7881	034276	000167	000560		JMP	501	; ABORT TEST, ERROR FOUND DURING DMA TX.
7882							
7883							
7884							
7885							
7886	034302	005267	147544		INC	ERRNBR	; SET ERROR NUMBER TO 6703.
7887	034306	012701	170536		MOV	#170536, R1	; PASS TIME-OUT VALUE OF 350 MILLI SECS.
7888	034312	016702	145710		MOV	CSRA, R2	; PASS THE ADDRESS OF THE CSR.
7889	034316	004767	163114		JSR	PC, WAITBIS	; WAIT FOR DMA TO COMPLETE, TX ACTION SET.
7890	034322	103402			BCS	.+6	; IF NO TIME-OUT ON DMA COMPLETION, DON'T ABORT.
7891	034324	000167	000532		JMP	501	; ABORT TEST, TIME-OUT ON DMA COMPLETION.
7892	034330	012704	000005		MOV	#5, R4	; PASS DELAY OF 5 MILLI SECS.
7893	034334	004767	157722		JSR	PC, DELAY	; WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
7894							
7895							
7896							
7897	034340	016705	145654		MOV	ACTLNS, R5	; ALTER PARAMETERS FOR ALL ACTIVE LINES.
7898	034344	012700	000204		MOV	#204, R0	; PASS PARAMETER FOR INTERNAL LOPBCK, ENABLE RX.
7899	034350	004767	163176		JSR	PC, WTM, NC	; INITIALISE THE LINE CONTROL REGISTER.
7900	034354	012700	177670		MOV	#177670, R0	; PASS THE LPR CONTENTS.
7901	034360	004767	163216		JSR	PC, WTM, LPR	; SET THE LPR CONTENTS TO 38.4K BAUD.
7902	034364	012704	000012		MOV	#10, R4	; PASS DELAY TIME OF 10 MILLI SECONDS.
7903	034370	004767	157666		JSR	PC, DELAY	; WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
7904							
7905	034374	012702	002650		MOV	#BUF BAS, R2	; PASS THE START OF THE DATA PATTERN TO TX.
7906	034400	012703	000005		MOV	#5, R3	; PASS THE LENGTH OF THE DATA PATTERN.
7907	034404	005001			CLR	R1	; CLEAR THE LINE COUNTER.
7908	034406	005267	147440		INC	ERRNBR	; SET ERROR NUMBER TO 6704.
7909	034412	010100			MOV	R1, R0	
7910	034414	006300			ASL	R0	; CALCULATE LINE OFFSET FROM THE LINE 0.
7911	034416	036067	002336	145574	BIT	BITBL(R0), ACTLNS	; TEST FOR THIS LINE BEING ACTIVE.
7912	034424	001405			BEQ	41	; SKIP THE TX ON THIS LINE IF IT IS NOT ACTIVE.
7913	034426	004767	157746		JSR	PC, DODMA	; TRANSMIT THE 5 CHAR DATA PATTERN.
7914	034432	103402			BCS	.+6	; IF NO TIME-OUT ON DMA COMPLETION, DON'T ABORT.
7915	034434	000167	000422		JMP	501	; ABORT TEST, TIME-OUT ON DMA COMPLETION.
7916	034440	005201			INC	R1	; INCREMENT THE LINE NUMBER COUNTER.
7917	034442	020127	000020		CMF	R1, #NUMLNS	; TEST FOR ALL POSSIBLE LINES HANDLED
7918	034446	002761			BLT	21	; LOOP IF NOT ALL LINES HANDLED.
7919							
7920	034450	005267	147376		INC	ERRNBR	; SET ERROR NUMBER TO 6705.
7921	034454	012701	170040		MOV	#170040, R1	; PASS TIME-OUT VALUE OF 32 MILLI SECS.
7922	034460	016702	145542		MOV	CSRA, R2	; PASS THE ADDRESS OF THE CSR.
7923	034464	004767	162746		JSR	PC, WAITBIS	; WAIT FOR A DMA TO COMPLETE, TX ACTION SET.
7924	034470	103174			BCC	501	; ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
7925	034472	012704	000005		MOV	#5, R4	; PASS DELAY OF 5 MILLI SECS.
7926	034476	004767	157560		JSR	PC, DELAY	; WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
7927							
7928							
7929							
7930							
7931	034502	012704	000400		MOV	#256, R4	; SET UP THE CHARACTER COUNTER.
7932	034506	012767	015062	147336	MOV	#6706, ERRNBR	; SET UP ERROR NUMBER EACH TIME AROUND LOOP.
7933	034514	017702	145510		MOV	#BUFA, R2	; READ A CHARACTER FROM THE FIFO.
7934	034520	100160			BPL	501	; ABORT THE TEST IF DATA, VALID IS CLEAR.

```

7935 034522 005267 147324      INC      ERRNBR      ;SET ERROR NUMBER TO 6707.
7936 034526 004767 157244      JSR      PC,CHKBMP  ;CHECK IF CHARACTER IS A BMP CODE.
7937 034532 103551              BLS      241      ;REPORT ERROR AND ABORT TEST IF A BMP CODE.
7938 034534 005304              DEC      R4          ;COUNT THIS CHARACTER.
7939 034536 001363              BNE      61          ;LOOP IF NOT 256 CHARS READ FROM FIFO.
7940
7941      ;
7942      ; READ THE REMAINING AND VERIFY 1 OVERRUN PLUS 1 CHAR FROM EACH LINE.
7943      ;
7943 034540 005004              CLR      R4          ;CLEAR THE OVERRUN ERROR FLAGS.
7944 034542 012700 003710      MOV      @RXCNTR,R0
7945 034546 004767 157466      JSR      PC,CLR16W  ;CLEAR RX CHAR COUNT TABLE.
7946 034552 012767 015064 147272 81:  MOV      @6708,ERRNBR  ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
7947 034560 017702 145444      MOV      @RBUF,R2      ;READ A CHARACTER FROM THE FIFO.
7948 034564 100047              BPL      141      ;GO ANALYZE THE RESULTS IF ALL CHARS READ.
7949 034566 004767 157204      JSR      PC,CHKBMP  ;CHECK IF CHAR IS A BMP CODE.
7950 034572 103531              BCS      241      ;REPORT ERROR AND ABORT TEST IF A BMP CODE.
7951 034574 005267 147252      INC      ERRNBR      ;SET ERROR NUMBER TO 6709.
7952 034600 010200              MOV      R2,R0
7953 034602 000300              SWAB     R0
7954 034604 042700 177760      BIC      @177760,R0  ;CALCULATE THE LINE NUMBER OF THE CHAR.
7955 034610 006300              ASL      R0          ;FORM WORD TABLE OFFSET FOR TABLE ACCESS.
7956 034612 042702 007400      BIC      @7400,R2      ;REMOVE LINE NUMBER FROM THE READ CHAR.
7957 034616 036067 002336 145374      BIT      BITTBL(R0),ACTLNS ;TEST FOR ACTIVE LINE.
7958 034624 001516              BEQ      501      ;ABORT TEST IF FOR INACTIVE LINE.
7959 034626 005267 147220      INC      ERRNBR      ;SET ERROR NUMBER TO 6710.
7960 034632 005760 003710      TST      RXCNTR(R0)  ;CHECK THE RX CHAR COUNTER FOR THIS LINE.
7961 034636 001006              BNE      101      ;IS THIS FIRST CHAR ON LINE?
7962 034640 020227 140000      CMP      R2,@140000  ;YES, TEST FOR NULL CHAR WITH OVERRUN.
7963 034644 001414              BEQ      121      ;IS CHAR A NULL?
7964 034646 056004 002336      BIS      BITTBL(R0),R4  ;NO, SET THE OVERRUN BIT ERROR FLAG FOR LINE.
7965 034652 000411              BR       121      ;GO COUNT THE CHAR AND CONTINUE.
7966 034654 026027 003710 000004 101:  CMP      RXCNTR(R0),@4
7967 034662 002077              BGE      501      ;5TH CHAR ON THIS LINE? YES, ABORT.
7968 034664 032702 040000      BIT      @BIT14,R2  ;NO, CHECK OVERRUN BIT.
7969 034670 001402              BEQ      121      ;IS OVERRUN BIT CLEAR? YES, GO COUNT CHAR.
7970 034672 056004 002336      BIS      BITTBL(R0),R4  ;NO, SET THE OVERRUN BIT ERROR FLAG FOR LINE.
7971 034676 005260 003710      INC      RXCNTR(R0)  ;COUNT THIS CHARACTER.
7972 034702 000723              BR       81          ;LOOP UNTIL ALL CHARS ARE READ FROM FIFO.
7973
7974      ;
7975      ; TEST FOR ABORT CONDITIONS. ONLY NONE ABORT CONDITIONS ARE:
7976      ; 1) 2 CHARS RXED ON A LINE AND NO OVERRUN ERROR BIT FAILURE DETECTED.
7977      ; 2) 2 TO 4 CHARS RXED ON A LINE AND AN OVERRUN BIT FAILURE DETECTED.
7978 034704 005001 141:  CLR      R1          ;INITIALIZE LINE LOOP, CLEAR LINE OFFSET.
7979 034706 012767 015067 147136 161:  MOV      @6711,ERRNBR  ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
7980 034714 036167 002336 145276      BIT      BITTBL(R1),ACTLNS
7981 034722 001415              BEQ      181      ;LINE ACTIVE? NO, NEXT LINE.
7982 034724 026127 003710 000002      CMP      RXCNTR(R1),@2  ;YES.
7983 034732 002453              BLT      501      ;FEWER THAN 2 CHARS RXED? YES, ABORT.
7984 034734 036104 002336      BIT      BITTBL(R1),R4  ;NO.
7985 034740 001006              BNE      181      ;OVERRUN BIT ERROR FLAG SET? YES, NEXT LINE.
7986 034742 005267 147104      INC      ERRNBR      ;SET LINE NUMBER TO 6712.
7987 034746 026127 003710 000002      CMP      RXCNTR(R1),@2
7988 034754 001042              BNE      501      ;NOT 2 CHARS RXED? YES, ABORT, NO, NEXT LINE.
7989 034756 062701 000002 181:  ADD      @2,R1          ;SET LINE OFFSET TO THE NEXT LINE.
7990 034762 020127 000040      CMP      R1,@NUM1HS+2
7991 034766 002747              BIT      161      ;ALL LINES DONE? NO, LOOP, YES, CONTINUE.

```

```

7992
7993
7994
7995 034770 012767 015071 147054
7996 034776 005001
7997 035000 010102
7998 035002 036104 002336
7999 035006 001415
8000 035010 010103
8001 035012 006203
8002 035014 012767 012506 147034
8003 035022 012701 010546
8004
8005 035026
      035026 104460
8006
8007
8008
8009
8010 035030 032767 000100 145152
8011 035036 001413
8012
8013
8014 035040 010201
8015 035042 046104 002336
8016 035046 001407
8017 035050 062701 000002
8018 035054 000751
8019
8020 035056
8021 035056
      035056 104460
8022 035060 000402
8023
8024 035062 004767 161604
8025 035066 005067 145156
8026
8027 035072
      035072
      035072 104401

; CHECK FOR OVERRUN ERROR BIT FAILURES, PRINT ERROR MESSAGE IF FOUND.
;
      MOV     #6715,ERRNBR ;SET UP ERROR NUMBER.
      CLR     R1           ;INITIALIZE LOOP. CLEAR LINE OFFSET.
20$:      MOV     R1,R2     ;COPY THE LINE OFFSET.
      BIT     BITTBL(R1),R4 ;OVERRUN BIT FAILURE FLAGS ARE IN R4.
      BEQ     22$         ;ERROR FLAG CLEAR? YES, NEXT LINE.
      MOV     R1,R3
      ASR     R3           ;CALCULATE LINE NUMBER FROM LINE OFFSET.
      MOV     #ERR7801,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
      MOV     #EM6702,R1   ;PASS THE MESSAGE TO BE REPORTED.
      REPORT  "OVERRUN ERROR NOT REPORTED CORRECTLY WHEN FORCED, ON LINE NN ..."
      ERROR
      ;
      ;>>>> ERROR #6713 <<<<<
      TRAP     C$ERROR

; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
;
      BIT     #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
      BEQ     60$           ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                           ;DURING THE SOFTWARE QUESTIONS.

      MOV     R2,R1
      BIC     BITTBL(R1),R4 ;RESTORE THE LINE OFFSET THAT WAS DESTROYED.
      BEQ     60$           ;CLEAR THE LINE ERROR FLAG WE JUST HANDLED.
      ADD     #2,R1         ;ALL FAILURE BITS HANDLED? YES, EXIT TEST.
      BR      20$          ;NO, INCREMENT THE LINE OFFSET.
                           ;LOOP TO HANDLE THE NEXT LINE.

24$:      REPORT  "BMP CODE FOUND IN FIFO, TEST INVAILDATED."
      ERROR
      ;
      ;>>>> ERROR <<<<<
      TRAP     C$ERROR

      BR      60$          ;EXIT THIS TEST.

50$:      JSR     PC,TSABRT ;ABORT THE TEST. ERROR # INDICATES FAULT TYPE.
60$:      CLR     CTRLCF   ;INDICATE THAT WE ARE NOT WITHIN A TEST.

      ENDIST

L10050:
      TRAP     C$EIST

```

```

8029 .SBTTL  HARDWARE TEST - REP BMP -
8030 *** *****
8031 ;* - REPORT ANY BMP CODES IN THE QUEUE -
8032 ;* THIS IS A PSEUDO-TEST USED TO REPORT ANY BMP CODES THAT WERE FOUND
8033 ;* IN THE DUT'S FIFO DURING PREVIOUS TEST, AND LOGGED IN THE BMP CODE
8034 ;* QUEUE.
8035 ;* IT IS UNLIKELY THAT RUNNING THIS PSEUDO-TEST ALONE WILL PRODUCE ANY
8036 ;* ERROR REPORTS.
8037 ;*
8038 ;* *****
8039 035074 BGNTST
      035074
8040 TNUM ** TNUM + 1 ; INCREMENT THE ASSEMBLY TIME TEST COUNTER.
8041 035074 012767 000024 145152 MOV #TNUM,TSTNUM ; SET UP THE TEST NUMBER. (93)
8042 035102 012767 177777 145140 MOV #-1,CTRLCF ; INDICATE THAT WE ARE IN A TEST.
8043 035110 016702 145274 MOV BMPCQP,R2 ; GET THE CONTENTS OF THE POINTER.
8044 035114 012703 002412 MOV #BMPCQB,R3 ; GET THE START ADDRESS OF THE QUEUE.
8045 035120 020203 CMP R2,R3 ; SEE IF THE POINTER HAS MOVED FROM THE BASE.
8046 035122 001411 BEQ 60$ ; EXIT NO CODES IN THE QUEUE.
8047 ;*
8048 ;* THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
8049 ;*
8050 ;REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN"
8051
8052 035124 012701 011354 MOV #EM9304,R1 ; PASS THE FIRST MESSAGE TO BE REPORTED.
8053 035130 012701 011354 ERDF 9301,EM9301,ER9301 ; ***** ERROR #9301 <<<<<.
      035130 104455 TRAP C$ERDF
      035132 022125 .WORD 9301
      035134 011200 .WORD EM9301
      035136 013176 .WORD ER9301
8054
8055 035140 012767 002412 145242 MOV #BMPCQB,BMPCQP ; SET POINTER BACK TO THE BEGINING OF THE QUE.
8056
8057 035146 005067 145076 60$: CLR CTRLCF ; INDICATE THAT WE ARE NOT WITHIN A TEST.
8058 035152 ENDTST
      035152 L10051: TRAP C$ETST
      035152 104401

```

8067
8068
8069
8070
8071
8072
8073
8074
8075
8076
8077
8078
8079
8080
8081

.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.
;--

```

8082 035154
035154 000022
035156

BGNHRD

.WORD L10052-L\$HARD/2
L\$HARD::8083
8093

;DEVICE CSR ADDRESS QUESTION:

8094 035156
035156 000031
035160 035222
035162 160000
035164 177776

GPRMA HWPTQ1,0,0,160000,177776,YES

.WORD T\$CODE
.WORD HWPTQ1
.WORD T\$LOLIM
.WORD T\$HILIM

8095

;DEVICE INTERRUPT VECTOR QUESTION:

8096 035166
035166 001031
035170 035240
035172 000040
035174 000776

GPRMA HWPTQ2,2,0,40,776,YES

.WORD T\$CODE
.WORD HWPTQ2
.WORD T\$LOLIM
.WORD T\$HILIM

8097

;ACTIVE LINES BIT MAP QUESTION:

8098 035176
035176 002032
035200 035273
035202 177777
035204 000000
035206 177777

GPRMD HWPTQ3,4,0,MAPLNS,0,177777,YES

.WORD T\$CODE
.WORD HWPTQ3
.WORD MAPLNS
.WORD T\$LOLIM
.WORD T\$HILIM

8099

;TYPE OF LOOPBACK QUESTION:

8100 035210
035210 003032
035212 035321
035214 000377
035216 000001
035220 000002

GPRMD HWPTQ4,6,0,377,1,2,YES

.WORD T\$CODE
.WORD HWPTQ4
.WORD 377
.WORD T\$LOLIM
.WORD T\$HILIM

8101

8102

8103 035222

ENDHRD

.EVEN

035222

L10052:

8104

8111

8112 035222 103 123 122
035225 040 101 104
035230 104 122 105
035233 123 123 072
035236 040 000

HWPTQ1: .ASCII /CSR ADDRESS: /

8113	035240	111	116	124	HWPTQ2: .ASCIZ /INTERRUPT VECTOR ADDRESS: /
	035243	105	122	122	
	035246	125	120	124	
	035251	040	126	105	
	035254	103	124	117	
	035257	122	040	101	
	035262	104	104	122	
	035265	105	123	123	
	035270	072	040	000	
8114	035273	101	103	124	HWPTQ3: .ASCIZ /ACTIVE LINE BIT MAP: /
	035276	111	126	105	
	035301	040	114	111	
	035304	116	105	040	
	035307	102	111	124	
	035312	040	115	101	
	035315	120	072	040	
	035320	000			
8115	035321	124	131	120	HWPTQ4: .ASCIZ /TYPE OF LOOPBACK (1=INTERNAL, 2=H3029 OR H3277):/
	035324	105	040	117	
	035327	106	040	114	
	035332	117	117	120	
	035335	102	101	103	
	035340	113	040	050	
	035343	061	075	111	
	035346	116	124	105	
	035351	122	116	101	
	035354	114	054	040	
	035357	062	075	110	
	035362	063	060	062	
	035365	071	040	117	
	035370	122	040	110	
	035373	063	062	067	
	035376	067	051	072	
	035401	000			
8116					
8117					.EVEN

```

8126
8127
8128 .SBTTL SOFTWARE PARAMETER CODING SECTION
8129
8130 ;**
8131 ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
8132 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
8133 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8134 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
8135 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
8136 ; WITH THE OPERATOR.
8137 ;**
8138
8139 035402          BGNSFT
      035402      000014
      035404
                                     .WORD L10053-L$SOFT/2
                                     L$SOFT::
8140
8149 ;UNIT NUMBER PRINTOUT QUESTION:
8150 035404          GPRML SWPTQ1,0,20,YES
      035404      000130
      035406      035434
      035410      000020
                                     .WORD T$CODE
                                     .WORD SWPTQ1
                                     .WORD 20
8151 ;EXTENDED ERROR REPORTING QUESTION:
8152 035412          GPRML SWPTQ2,0,100,YES
      035412      000130
      035414      035510
      035416      000100
                                     .WORD T$CODE
                                     .WORD SWPTQ2
                                     .WORD 100
8153
8154 ; IF EXTENDED ERROR REPORTING IS NOT REQUIRED THEN SKIP THE NEXT QUESTION.
8155 ;**
8156 035420          XFERF ENDD
      035420      006044
                                     .WORD T$CODE
8157
8158 ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
8159 035422          GPRMD SWPTQ3,2,D,177777,0,177777,YES
      035422      001052
      035424      035543
      035426      177777
      035430      000000
      035432      177777
                                     .WORD T$CODE
                                     .WORD SWPTQ3
                                     .WORD 177777
                                     .WORD T$LOLIM
                                     .WORD T$HILIM
8160
8161          .EVEN
8162
8163 035434          ENDD: ENDSFT
      035434
                                     .EVEN
                                     L10053:
8164
8165
8172 035434      122      105      120      SWPTQ1: .ASCIIZ /REPORT UNIT NUMBER AS EACH UNIT IS TESTED:
      035437      117      122      124
      035442      040      125      116
      035445      111      124      040
      035450      116      125      115
      035453      102      105      122
      035456      040      101      123
      035461      040      105      101
      035464      103      110      040

```

	035467	125	116	111	
	035472	124	040	111	
	035475	123	040	124	
	035500	105	123	124	
	035503	105	104	072	
	035506	040	000		
8173	035510	105	130	124	SWPTQ2: .ASCIZ /EXTENDED ERROR REPORTING: /
	035513	105	116	104	
	035516	105	104	040	
	035521	105	122	122	
	035524	117	122	040	
	035527	122	105	120	
	035532	117	122	124	
	035535	111	116	107	
	035540	072	040	000	
8174	035543	116	125	115	SWPTQ3: .ASCIZ /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /
	035546	102	105	122	
	035551	040	117	106	
	035554	040	111	116	
	035557	104	111	126	
	035562	111	104	125	
	035565	101	114	040	
	035570	104	101	124	
	035573	101	040	105	
	035576	122	122	117	
	035601	122	123	040	
	035604	124	117	040	
	035607	122	105	120	
	035612	117	122	124	
	035615	040	117	116	
	035620	040	101	040	
	035623	114	111	116	
	035626	105	072	040	
8175	035631	000			.EVEN

```
8184
8185
8186 035632          $PATCH::
8187 035632          .BLKW    24
8188
8195
8196
8197
8198
8199 035702          LASTAD
                                .EVEN
                                .WORD    0
                                .WORD    0
                                035702 000000
                                035704 000000
                                035706
8200 035706          L$LAST::
                                ENDMOD
8201
8202
8203
8204
8205
8206
8207
8208          000001          .END
```

ACTLNS	002220	G	CTRLCF	002250	G	DMABUF	014322	G	EM5804	007201	G	FDATE	-	000006	G
ADR	000020	G	C\$AU	000052		DODMA	014400	G	EM5805	007265	G	FINACT	014470	G	
ADRPTR	014200	G	C\$AUTO	000061		DRADRT	002226	G	EM5901	007346	G	FSLSA	002234	G	
ALTFLD	013370	G	C\$BRK	000022		DROP	020730		EM5902	007402	G	FSLSO	000006	G	
ASLNTL	013442	G	C\$BSEG	000004		EDROP	021006		EM5903	007476	G	F\$AU	000015		
ASSEMB	000010		C\$BSUB	000002		EF.CON	000036	G	EM5904	007570	G	F\$AUTO	000020		
BCOUN1	002322	G	C\$CEFG	000045		EF.NEW	000035	G	EM5905	007654	G	F\$BGN	000040		
BITBL	002336	G	C\$CLCK	000062		EF.PWR	000034	G	EM6001	007754	G	F\$CLEA	000007		
BIT0	000001	G	C\$CLEA	000012		EF.RES	000037	G	EM6002	010000	G	F\$DU	000016		
BIT00	000001	G	C\$CLOS	000035		EF.STA	000040	G	EM6003	010040	G	F\$END	000041		
BIT01	000002	G	C\$CLP1	000006		EF0503	004156	G	EM6004	010120	G	F\$HARD	000004		
BIT02	000004	G	C\$CVEC	000036		EF1601	004163	G	EM6005	010165	G	F\$HW	000013		
BIT03	000010	G	C\$DCLN	000044		EF5801	004215	G	EM6006	010243	G	F\$INIT	000006		
BIT04	000020	G	C\$DODU	000051		EF5901	004270	G	EM6401	010307	G	F\$JMP	000050		
BIT05	000040	G	C\$DRPT	000024		EF5902	004320	G	EM6402	010344	G	F\$MOD	000000		
BIT06	000100	G	C\$DU	000053		EF6401	004350	G	EM6601	010405	G	F\$MSG	000011		
BIT07	000200	G	C\$EDIT	000003		EF7801	004417	G	EM6602	010442	G	F\$PROT	000021		
BIT08	000000	G	C\$ERDF	000055		EF9001	004455	G	EM6701	010514	G	F\$PWR	000017		
BIT09	001000	G	C\$ERMR	000056		EF9002	004537	G	EM6702	010546	G	F\$RPT	000012		
BIT1	000002	G	C\$ERRO	000060		EF9003	004611	G	EM9009	010623	G	F\$SEG	000003		
BIT10	002000	G	C\$ERSF	000054		EF9004	004640	G	EM9010	010647	G	F\$SOFT	000005		
BIT11	004000	G	C\$ERSO	000057		EF9005	004670	G	EM9014	010673	G	F\$SRV	000010		
BIT12	010000	G	C\$ESCA	000010		EF9006	004721	G	EM9017	010767	G	F\$SUB	000002		
BIT13	020000	G	C\$ESEG	000005		EF9010	004745	G	EM9026	011100	G	F\$SW	000014		
BIT14	040000	G	C\$ESUB	000003		EF9019	005034	G	EM9104	011124	G	F\$TEST	000001		
BIT15	100000	G	C\$ETST	000001		EF9301	005053	G	EM9301	011200	G	GETPRM	020446		
BIT2	000004	G	C\$EXIT	000032		EF9302	005131	G	EM9302	011257	G	GPRSOB	002376	G	
BIT3	000010	G	C\$GETB	000026		EM0101	015124	G	EM9303	011307	G	G\$CNTD	000200		
BIT4	000020	G	C\$GETW	000027		EM0102	015210	G	EM9304	011354	G	G\$DELM	000372		
BIT5	000040	G	C\$GMAN	000043		EM0103	005262	G	ENDD	035434		G\$DISP	000003		
BIT6	000100	G	C\$GPHR	000042		EM1601	005320	G	ENDET8	003650	G	G\$EXCP	000400		
BIT7	000200	G	C\$GPLO	000030		EM4001	005403	G	ENDIT	020650		G\$HILI	000002		
BIT8	000400	G	C\$GPRI	000040		EM4002	005435	G	ERCNT8	002614	G	CLI	000001		
BIT9	001000	G	C\$INIT	000011		EM4101	005471	G	ERLTBL	002650	G	G\$NO	000000		
BMPCQB	002412	G	C\$INLP	000020		EM4102	005523	G	ERRBLK	004056	G	G\$CFFS	000400		
BMPCQE	002612	G	C\$MANI	000050		EM4103	005557	G	ERRMSG	004054	G	G\$OF SI	000376		
BMPCQP	002410	G	C\$MEM	000031		EM4201	005643	G	ERRNBR	004052	G	G\$PRMA	000001		
BOE	000400	G	C\$MSG	000023		EM4202	005675	G	ERRTYP	004050	G	G\$PRMD	000002		
BRLEVL	002223	G	C\$OPEN	000034		EM4901	005717	G	ERSMRF	002612	G	G\$PEML	000000		
BUFBAS	002650	G	C\$PNTB	000014		EM4902	005760	G	ER0101	011430	G	G\$RADA	000140		
BUFEND	003650	G	C\$PNTF	000017		EM5001	006012	G	ER0503	011762	G	G\$RADB	000000		
BUFMID	003250	G	C\$PNTS	000016		EM5101	006051	G	ER1603	012020	G	G\$RADD	000040		
BUFPTR	002246	G	C\$PNTX	000015		EM5102	006106	G	ER5801	012112	G	G\$RADL	000120		
BUF3QT	003450	G	C\$QIO	000377		EM5103	006144	G	ER5901	012170	G	G\$RADO	000020		
CALMSL	013552	G	C\$RDBU	000007		EM5201	006174	G	ER6001	012276	G	G\$XFER	000004		
CHKBMP	013776	G	C\$REFG	000047		EM5202	006227	G	ER6401	012400	G	G\$YES	000010		
CKRXTH	014046	G	C\$RESE	000033		EM5301	006265	G	ER7801	012506	G	HELP	000000		
CKTRAP	014166	G	C\$REVI	000003		EM5302	006321	G	ER9001	012544	G	HOE	100000	G	
CLKBRL	002306	G	C\$RFLA	000021		EM5303	006401	G	ER9002	012644	G	HWPTQ1	035222		
CLKCSR	002304	G	C\$RPT	000025		EM5401	006452	G	ER9004	013022	G	HWPTQ2	035240		
CLKHRZ	002312	G	C\$SEFG	000046		EM5402	006520	G	ER9101	013136	G	HWPTQ3	035273		
CLKINT	017632	G	C\$SPRI	000041		EM5501	006561	G	ER9301	013176	G	HWPTQ4	035321		
CLKJEC	002310	G	C\$SVEC	000037		EM5601	006625	G	EVL	000004	G	IBL	010000	G	
CLKRST	014216	G	C\$TPRI	000013		EM5701	006702	G	EXDERR	002252	G	IDU	000040	G	
CLR16W	014240	G	DELAY	014262	G	EM5801	006757	G	E\$END	002100		IFR	020000	G	
CLRA	002226	G	DEPTBL	002176	G	EM5802	007003	G	E\$LOAD	000035		IFSTAT	002256	G	
CSRU	000000	G	DIAGMC	000000		EM5803	007077	G	FDATE	002234	G	INDATP	014550	G	

INDTPX 014600 G	L\$EXP4 002064 G	L10034 024714	PRI06 = 000300 G	TXBFCA 002244 G
ISR = 000100 G	L\$EXP5 002066 G	L10035 025406	PRI07 = 000340 G	TXBFCA = 000016 G
IXE = 004000 G	L\$HARD 035156 G	L10036 025714	PRTLPR 015306 G	TXDATP 017004 G
I\$AU = 000041	L\$HIME 002120 G	L10037 026224	PUFIFO 015370 G	TXDSBL 017026 G
I\$AUTO = 000041	L\$HPCP 002016 G	L10040 026724	PUFIFR 015452 G	TXENBL 017122 G
I\$CLN = 000041	L\$HPTP 002022 G	L10041 027422	RBUFA = 002230 G	TXIE0 017216 G
I\$DU = 000041	L\$HW 002176 G	L10042 030100	RBUFO = 000002 G	TXIE1 017256 G
I\$HRD = 000041	L\$ICP 002104 G	L10043 031150	READBX 015654 G	TXINTC 002272 G
I\$INIT = 000041	L\$INIT 020044 G	L10044 031736	REPDER 015736 G	TXINTF 002274 G
I\$MOD = 000041	L\$LADP 002026 G	L10045 032764	REPSMR 016254 G	TXRLNB 004010 G
I\$MSG = 000041	L\$LAST 035706 G	L10046 033410	RESETT 016302 G	TXRLNE 004030 G
I\$PROT = 000040	L\$LOAD 002100 G	L10047 034126	RXBOTX = 000030 G	TXRXLB 003750 G
I\$PTAB = 000041	L\$LUN 002074 G	L10050 035072	RXBETX = 000020 G	TXRXLE 004010 G
I\$PWR = 000041	L\$MREV 002050 G	L10051 035152	RXBFUL = 000100 G	TXVECA 002216 G
I\$RPT = 000041	L\$NAME 002000 G	L10052 035222	RXCNTB 003710 G	T\$ARGC = 000002
I\$SEG = 000041	L\$PRIO 002042 G	L10053 035434	RXDECT 017702 G	T\$CODE = 001052
I\$SETU = 000041	L\$PROT 020036 G	MAPCNT 014676 G	RXIE0 016414 G	T\$ERRN = 022125
I\$SFT = 000041	L\$PRT 002112 G	MAPLNS = 177777 G	RXIE1 016454 G	T\$EXCP = 000000
I\$SRV = 000041	L\$REPP 002062 G	MFUNIT 005231 G	RXINTC 002266 G	T\$FLAG = 000040
I\$SUB = 000041	L\$REV 002010 G	MMENAB 002334 G	RXINTF 002270 G	T\$GMAN = 000000
I\$TST = 000041	L\$RPT 020030 G	MMPRES 002332 G	RXTIMO = 000002 G	T\$HILI = 177777
J\$JMP = 000167	L\$SOFT 035404 G	MMSRO 002330 G	RXTMA 002230 G	T\$LAST = 000001
LGRP1M 002260 G	L\$SPC 002056 G	MSG1 011546 G	RXVECA 002214 G	T\$LOLI = 000000
LGRP2M 002262 G	L\$SPCP 002020 G	MSG2 011624 G	R0SLOT = 000002 G	T\$LSYM = 010000
LINBIT 014650 G	L\$SPTP 002024 G	MSG3 011703 G	R1SLOT = 000004 G	T\$LTNO = 000024
LNCTRA 002236 G	L\$STA 002030 G	MSLCNT 002326 G	R2SLOT = 000006 G	T\$NEST = 177777
LNCTRO = 000010 G	L\$SW 002210 G	MSLGET 014730 G	R3SLOT = 000010 G	T\$NS0 = 000000
LOE = 040000 G	L\$TEST 002114 G	MSLOOP 015044 G	R4SLOT = 000012 G	T\$NS1 = 000005
LOPCK 002222 G	L\$TIML 002014 G	MSTICK 002324 G	R5SLOT = 000014 G	T\$PINU = 000000
LOT = 000010 G	L\$UNIT 002012 G	NDERPT 002212 G	SAVBMP 016500 G	T\$SAVL = 177777
LPCSLT = 000036 G	L10000 002206	NEWPAS 020426	SETPAR 016546 G	T\$SEGL = 177777
LPRA = 002232 G	L10001 002214	NEWRES 020420	SFPTBL 002210 G	T\$SUBN = 000000
LPRO = 000004 G	L10002 011544	NEWSTA 020110	SKPSTS 016614 G	T\$TAGL = 177777
L\$ACP 002110 G	L10003 012016	NUMLNS = 000020 G	STGTRB 004030 G	T\$TAGN = 010054
L\$APR 002036 G	L10004 012110	OOPS 015060 G	SVCGBL = 000000	T\$TEMP = 000000
L\$AU 021014 G	L10005 012166	OPTION 002210 G	SVCINS = 000001	T\$TEST = 000024
L\$AUT 002070 G	L10006 012274	O\$APTS = 000000	SVCSUB = 000001	T\$TSTM = 177777
L\$AUTO 020664 G	L10007 012376	O\$AU = 000000	SVCTAG = 000001	T\$TSTS = 000001
L\$CCP 002106 G	L10010 012504	O\$BGNR = 000001	SVCTST = 000001	T\$TAU = 010025
L\$CLEA 020666 G	L10011 012542	O\$BGNS = 000001	SWPTQ1 035434	T\$TAUT = 010022
L\$CO 002032 G	L10012 012642	O\$DU = 000001	SWPTQ2 035510	T\$TCLE = 010023
L\$DEPO 002011 G	L10013 013020	O\$ERRT = 000001	SWPTQ3 035543	T\$TDU = 010024
L\$DESC 004130 G	L10014 013134	O\$GNSW = 000001	S\$LSYM = 010000	T\$THAR = 010052
L\$DESP 002076 G	L10015 013174	O\$POIN = 000001	TIMER1 002314 G	T\$THW = 010000
L\$DEVP 002060 G	L10016 013366	O\$SETU = 000000	TIMER2 002316 G	T\$TINT = 010021
L\$DISP 002124 G	L10017 020034	PASCNT 002264 G	TIMER3 002320 G	T\$THSG = 010016
L\$DLY 002116 G	L10021 020662	PCSLT = 000016 G	TNUM = 000024 G	T\$THPRO = 010020
L\$DTP 002040 G	L10022 020664	PNT = 001000 G	TP4FLG 002300 G	T\$THRPT = 010017
L\$DTYP 002034 G	L10023 020702	PREGRT 004102 G	TP4RTN 017712 G	T\$THSO = 010053
L\$DU 020704 G	L10024 021012	PREGOS 004060	TP4VEC 002276 G	T\$THSW = 010001
L\$DUT 002072 G	L10025 021020	PRI = 002000 G	TSABRT 016672 G	T\$THTES = 010051
L\$DVTY 004120 G	L10026 021302	PRI00 = 000000 G	TSTNUM 002254 G	T1 021022 G
L\$EF 002052 G	L10027 021706	PRI01 = 000040 G	TXAD1A 002240 G	T10 025716 G
L\$ENVI 002044 G	L10030 022344	PRI02 = 000100 G	TXAD10 = 000012 G	T11 026226 G
L\$ERRT 004050 G	L10031 022672	PRI03 = 000140 G	TXAD2A 002242 G	T12 026726 G
L\$ETP 002102 G	L10032 023472	PRI04 = 000200 G	TXAD20 = 000014 G	T13 027424 G
L\$EXP1 002046 G	L10033 024272	PRI05 = 000240 G	TXAINT 017734 G	T14 030102 G

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	GLOBAL SUBROUTINE	I9	HARDWARE TEST	I13	
PROGRAM DOCUMENT	J1	GLOBAL ERROR REPORTI....J5	GLOBAL SUBROUTINE	J9	HARDWARE TEST	J13	
PROGRAM DOCUMENT	K1	GLOBAL ERROR REPORTI....K5	GLOBAL SUBROUTINE	K9	HARDWARE TEST	K13	
PROGRAM DOCUMENT	L1	GLOBAL ERROR REPORTI....L5	GLOBAL SUBROUTINE	L9	HARDWARE TEST	L13	
PROGRAM DOCUMENT	M1	GLOBAL ERROR REPORTI....M5	GLOBAL SUBROUTINE	M9	HARDWARE TEST	M13	
PROGRAM DOCUMENT	N1	GLOBAL SUBROUTINES S....N5	GLOBAL SUBROUTINE	N9	HARDWARE TEST	N13	
PROGRAM DOCUMENT	B2	GLOBAL SUBROUTINE	B6	GLOBAL SUBROUTINE	B10	HARDWARE TEST	B14
PROGRAM DOCUMENT	C2	GLOBAL SUBROUTINE	C6	GLOBAL SUBROUTINE	C10	HARDWARE TEST	C14
PROGRAM DOCUMENT	D2	GLOBAL SUBROUTINE	D6	INTERRUPT SERVICE RO....D10	HARDWARE TEST	D14	
PROGRAM DOCUMENT	E2	GLOBAL SUBROUTINE	E6	INTERRUPT SERVICE RO....E10	HARDWARE TEST	E14	
PROGRAM DOCUMENT	F2	GLOBAL SUBROUTINE	F6	GLOBAL TRAP SERVICEF10	HARDWARE TEST	F14	
PROGRAM DOCUMENT	G2	GLOBAL SUBROUTINE	G6	INTERRUPT SERVICE RO....G10	HARDWARE TEST	G14	
PROGRAM DOCUMENT	H2	GLOBAL SUBROUTINE	H6	INTERRUPT SERVICE RO....H10	HARDWARE TEST	H14	
PROGRAM DOCUMENT	I2	GLOBAL SUBROUTINE	I6	PROTECTION TABLEI10	HARDWARE TEST	I14	
PROGRAM DOCUMENT	J2	GLOBAL SUBROUTINE	J6	PROTECTION TABLEJ10	HARDWARE TEST	J14	
PROGRAM DOCUMENT	K2	GLOBAL SUBROUTINE	K6	INITIALIZE SECTIONK10	HARDWARE TEST	K14	
PROGRAM DOCUMENT	L2	GLOBAL SUBROUTINE	L6	INITIALIZE SECTIONL10	HARDWARE TEST	L14	
PROGRAM HEADER	M2	GLOBAL SUBROUTINE	M6	INITIALIZE SECTIONM10	HARDWARE TEST	M14	
PROGRAM HEADER	N2	GLOBAL SUBROUTINE	N6	INITIALIZE SECTIONN10	HARDWARE TEST	N14	
DISPATCH TABLE	E7	GLOBAL SUBROUTINE	B7	AUTODROP SECTION	B11	HARDWARE TEST	B15
DISPATCH TABLE	C7	GLOBAL SUBROUTINE	C7	CLEANUP CODING SECTI....C11	HARDWARE TEST	C15	
DEFAULT HARDWARE P-T....D3	GLOBAL SUBROUTINE	D7	DROP UNIT SECTION	D11	HARDWARE TEST	D15	
SOFTWARE P-TABLE	E3	GLOBAL SUBROUTINE	E7	HARDWARE TEST	E11	HARDWARE TEST	E15
GLOBAL EQUATES SECTI....F3	GLOBAL SUBROUTINE	F7	HARDWARE TEST	F11	HARDWARE TEST	F15	
GLOBAL EQUATES SECTI....G3	GLOBAL SUBROUTINE	G7	HARDWARE TEST	G11	HARDWARE TEST	G15	
GLOBAL DATA SECTIONH3	GLOBAL SUBROUTINE	H7	HARDWARE TEST	H11	HARDWARE TEST	H15	
GLOBAL DATA SECTIONI3	GLOBAL SUBROUTINE	I7	HARDWARE TEST	I11	HARDWARE TEST	I15	
GLOBAL DATA SECTIONJ3	GLOBAL SUBROUTINE	J7	HARDWARE TEST	J11	HARDWARE TEST	J15	
GPR HANDLING ROUTINE....K3	GLOBAL SUBROUTINE	K7	HARDWARE TEST	K11	HARDWARE TEST	K15	
GPR FRAME ACCESS EQU....L3	GLOBAL SUBROUTINE	L7	HARDWARE TEST	L11	HARDWARE TEST	L15	
GLOBAL MACRO DEFINIT....M3	GLOBAL SUBROUTINE	M7	HARDWARE TEST	M11	HARDWARE TEST	M15	
GLOBAL MACRO DEFINIT....N3	GLOBAL SUBROUTINE	N7	HARDWARE TEST	N11	HARDWARE TEST	N15	
GLOBAL SUBROUTINE	B4	GLOBAL SUBROUTINE	B8	HARDWARE TEST	B12	HARDWARE TEST	B16
GLOBAL TEXT SECTIONC4	GLOBAL SUBROUTINE	C8	HARDWARE TEST	C12	HARDWARE TEST	C16	
GLOBAL TEXT SECTIOND4	GLOBAL SUBROUTINE	D8	HARDWARE TEST	D12	HARDWARE TEST	D16	
GLOBAL TEXT SECTIONE4	GLOBAL SUBROUTINE	E8	HARDWARE TEST	E12	HARDWARE TEST	E16	
GLOBAL TEXT SECTIONF4	GLOBAL SUBROUTINE	F8	HARDWARE TEST	F12	HARDWARE TEST	F16	
GLOBAL TEXT SECTIONG4	GLOBAL SUBROUTINE	G8	HARDWARE TEST	G12	HARDWARE TEST	G16	
GLOBAL ERROR REPORTI....H4	GLOBAL SUBROUTINE	H8	HARDWARE TEST	H12	HARDWARE PARAMETER C....H16		
GLOBAL ERROR REPORTI....I4	GLOBAL SUBROUTINE	I8	HARDWARE TEST	I12	HARDWARE PARAMETER C....I16		
GLOBAL ERROR REPORTI....J4	GLOBAL SUBROUTINE	J8	HARDWARE TEST	J12	SOFTWARE PARAMETER C....J16		
GLOBAL ERROR REPORTI....K4	GLOBAL SUBROUTINE	K8	HARDWARE TEST	K12	SOFTWARE PARAMETER C....K16		
GLOBAL ERROR REPORTI....L4	GLOBAL SUBROUTINE	L8	HARDWARE TEST	L12	SYMBOL TABLE	L16	
GLOBAL ERROR REPORTI....M4	GLOBAL SUBROUTINE	M8	HARDWARE TEST	M12	SYMBOL TABLE	M16	
GLOBAL ERROR REPORTI....N4	GLOBAL SUBROUTINE	N8	HARDWARE TEST	N12			

B1

CZDHV80.DMU-11 FUNC TST PART2 MACRO M1200 15-MAR 84 09:28 PAGE 127-3
SYMBOL TABLE

SEQ 208

T15	031152 G	T20	035074 G	T8	024716 G	WAIBIS	017436 G	X\$ALWA=	000000
T16	031740 G	T3	021710 G	T9	025410 G	WAITTX	017512 G	X\$FALS=	000040
T17	032766 G	T4	022346 G	UAM	000200 G	WORD1	002302 G	X\$OFFS=	000400
T18	033412 G	T5	022674 G	UNITN	002224 G	WTWLNC	017552 G	X\$TRUE=	000020
T19	034130 G	T6	023474 G	UNSDIV	017302 G	WTWLPR	017602 G	\$PATCH	035632 G
T2	021304 G	T7	024274 G						

.ABS. 000000 000
000000 001

ERRORS COLLECTED: 0

VIRTUAL MEMORY USED: 28661 WORDS (112 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:05:03
CZDHV80.BIN,CZDHV80.LST/-SP=SVC34R/ML,CZDHV80.P11

SYMBOL TABLE 81