

DHU11	DHU-11 FUNC TST PART3
	CZDHWB0

DHU-11 FUNC TST PART3
CZDHWB0

CZDHWO

COPYRIGHT (c) 1983-84

AH-T799B-MC

FICHE 01 OF 01

JUL 1984

digital

Made In USA.

.REM 6

IDENTIFICATION

PRODUCT CODE: AC-T798B MC
PRODUCT NAME: CZDHWB0 DMU-11 FUNC TST PART3
PRODUCT DATE: 3 MARCH 1984
MAINTAINER: ENE - DIAGNOSTICS GROUP
AUTHOR: ANTHONY HART
MODIFIED BY: ANTHONY HART

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF
SOFTWARE 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

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

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

ORIGINAL RELEASE:	15-DEC 1983	ANTHONY HART
VERSION B0	3-MAR-1984	ANTHONY HART

THE FOLLOWING MODIFICATIONS HAVE BEEN MADE TO THE OLD CZDHWA:

THE HARDWARE QUESTION "TYPE OF LOOPBACK" HAS BEEN ALTERED TO INCLUDE THE STAGGERED LOOPBACK CONNECTORS ON THE DMU11 DISTRIBUTION PANEL (H3029).

THE HARDWARE QUESTION "INTERRUPT VECTOR" HAS BEEN REMOVED.

THE HARDWARE QUESTION "BR LEVEL" HAS BEEN REMOVED.

ALL THE TESTS THAT WERE IN THE PREVIOUS VERSION (CZDHWA); EXCEPT THE "REGISTER ADDRESS TEST" AND THE "REPORT BMP CODES" (THE FIRST AND LAST TESTS COMMON TO EACH PART OF THE DIAGNOSTIC); HAVE BEEN TRANSFERED TO PART (CZDHX).

THE MODEM SIGNAL TESTS FROM PART CZDHVA HAVE BEEN TRANSFERED INTO THIS PART. THEY WERE TESTS 16 THRU 23 IN CZDHVA.

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

CZDMW80 IS PART OF THE DHU 11 FUNCTIONAL VERIFICATION TEST. THIS PART OF THE TEST PERFORMS TESTS ON THE MODEM CONTROL SIGNALS OF THE DUT.

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

1.2 SYSTEM REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DHU11 FVT:

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

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE PROCESSOR, THE UNIBUS, THE SYSTEM MEMORY, THE CONSOLE TERMINAL AND THE LOAD MEDIA ARE ASSUMED TO HAVE BEEN TESTED AND FOUND WORKING BEFORE THIS PROGRAM IS RUN.

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES.
FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL (CHQUS).

2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES
(SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY
BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
-----	-----
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER +C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SEE PERFORMANCE AND PROGRESS REPORTS SECTION OF THIS DOCUMENT)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE FLAGS SECTION)
ZFLAGS	CLEAR ALL FLAGS (SEE FLAGS SECTION)

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

2.2 SWITCHES

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

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS. FOR EXAMPLE - /TESTS:1:5:7 10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. SEE THE FLAGS SECTION OF THIS DOCUMENT.
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE /UNITS:0:5:10 12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0 63)

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

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

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

*SEE THE ERROR INFORMATION SECTION OF THIS DOCUMENT.

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

/FLAGS:LOE:IER:BOE

2.4 EXTENDED COMMAND SYNTAX

2.4.1 START COMMAND

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

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

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

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

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

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

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

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

IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC.
(HAS NO EFFECT IN THIS DIAGNOSTIC.)

LOT LOOP ON TEST.

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE
Cleared. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT
GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF "EFFECT OF START
COMMAND" SECTION.

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

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

2.4.1.5 EFFECT OF START COMMAND -

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

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION "4
UNITS (D) ?" TO WHICH THE OPERATOR SHOULD REPLY WITH THE NUMBER OF
UNITS TO BE TESTED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE
P-TABLES THEMSELVES ARE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE
CONTAINING ALL THE HARDWARE INFORMATION FOR ONE COMPLETE UNIT. EACH
QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR
BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT
VALUE AFTER THE PARENTHESES. FOR THE ACTUAL HARDWARE P TABLE
QUESTIONS SEE THE "HARDWARE PARAMETERS" SECTION.

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO
BUILD THE SOFTWARE TABLES, WHICH DEFINE OPERATING PARAMETERS OF THE
DIAGNOSTIC PROGRAM. THESE QUESTIONS ARE DESCRIBED IN THE "SOFTWARE
PARAMETERS" SECTION.

EXAMPLE:

STA/TESTS:1:3-4:/PASS:3/FLAGS:IER:HOE~1

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, WITH EACH PASS
CONSISTING OF TESTS 1,3, AND 4. THERE IS NO DIFFERENCE BETWEEN SAYING
<FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY
ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET.
NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

2.4.2 RESTART COMMAND -

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

2.4.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

2.4.2.3 EFFECT OF RESTART COMMAND -

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

2.4.3 CONTINUE COMMAND -

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

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

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

2.4.3.2 EFFECT OF CONTINUE COMMAND

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

2.4.4 PROCEED COMMAND -

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

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

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

2.4.4.2 EFFECT OF PROCEED COMMAND

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

2.4.5 ADD COMMAND

ADD/UNITS:<UNIT-LIST>

2.4.6 EFFECT OF ADD COMMAND -

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

2.4.7 DROP COMMAND

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

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

2.4.9 PRINT COMMAND -

PRI(N1)

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

2.4.10 DISPLAY COMMAND -

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

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

2.4.11 FLAGS COMMAND

FLA(GS)

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

2.4.12 ZFLAGS COMMAND -

ZFL(AGS)

2.4.13 ZFLAGS COMMAND -

ALL FLAGS ARE CLEARED.

2.4.14 CONTROL CHARACTERS

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

2.5 HARDWARE QUESTIONS

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

1. CSR ADDRESS - THIS QUESTION REQUESTS THE CSR ADDRESS OF THE SPECIFIED DMU-11. THE DEFAULT ANSWER FOR THIS QUESTION IS ADDRESS 160460 (OCTAL).
2. ACTIVE LINES BIT MAP - THIS QUESTION REQUESTS AN OCTAL BIT MAP OF THE SERIAL COMMUNICATION LINES ON THE DMU11 WHICH ARE BEING SELECTED FOR TESTING. IF THE BIT IN THE BIT MAP IS SET WHICH CORRESPONDS TO A PARTICULAR LINE (I.E. BIT 5 FOR LINE 5) THAT LINE WILL BE TESTED BY THE FVT. THE DEFAULT ANSWER FOR THIS QUESTION IS ALL LINES I.E. 177777.
3. TYPE OF LOOPBACK (1=INTERNAL, 2=H3029 OR H3277, 3=H325)
THIS QUESTION REQUESTS THE TYPE OF LOOPBACK TO BE USED WHEN TESTING THE DMU-11.
THE FOLLOWING TYPES ARE SUPPORTED:
 - 0 INTERNAL - ONLY INTERNAL UART LOOPBACK IS TO BE USED IN TESTING THE DMU-11. SINCE ALL THE TESTS IN THIS PART REQUIRE EITHER STAGGERED OR SINGLE LINE LOOPBACK, SPECIFYING INTERNAL LOOPBACK WILL CAUSE THE TESTS TO BE SKIPPED. THIS WILL NOT HOWEVER CAUSE ANY ERRORS TO BE REPORTED.
 - 0 H3029 OR H3277 - STAGGERED LOOPBACK CONNECTORS ARE PROVIDED ON THE DMU11 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 DMU11.
 - 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.

* UNITS (0) ? 8<CR>

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

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

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

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

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

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

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

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

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

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

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

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

```
UNIT 1
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 0.1<CR>
Q-FACTOR (0) 0 ? 1.0<CR>
```

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

```
UNIT 7
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 6.7<CR>
Q-FACTOR (0) 0 ? 1<CR>
```

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

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

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

UNIT 1
CSR ADDRESS (0) ? 1600C0<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:

CZDHW DVC FTL ERR 7802 ON UNIT 00 TST 002 SUB 000 PC: XXXXXX
MODEM CONTROL DTR BIT TEST FAILED

THIS ERROR INDICATES THAT A FATAL ERROR WAS ENCOUNTERED DURING THE TEST WHICH TESTS THE DATA TERMINAL READY CONTROL SIGNAL.

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

CZDHW DVC FTL ERR 7802 ON UNIT 00 TST 002 SUB 000 PC: XXXXXX
MODEM CONTROL DTR BIT TEST FAILED
DTR BIT FAULTY ON LINE 4 DECIMAL.

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 CZDMWB:

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. DTR TEST - VERIFIES THAT CHANGING THE STATE OF THE DTR BIT AFFECTS THE STATE OF THE DTR CONTROL LINE.
3. RTS TEST - VERIFIES THAT CHANGING THE STATE OF THE RTS BIT AFFECTS THE STATE OF THE RTS CONTROL LINE.
4. DSR TEST - VERIFIES THAT THE DSR STATUS SIGNAL CORRECTLY REPORTS THE STATE OF THE LOOPED BACK DTR CONTROL LINE.
5. RI TEST - VERIFIES THAT THE RI STATUS SIGNAL CORRECTLY REPORTS THE STATE OF THE LOOPED BACK DTR CONTROL LINE.
6. CTS TEST - VERIFIES THAT THE CTS STATUS SIGNAL CORRECTLY REPORTS THE STATE OF THE LOOPED BACK RTS CONTROL LINE.
7. DCD TEST - VERIFIES THAT THE DCD STATUS SIGNAL CORRECTLY REPORTS THE STATE OF THE LOOPED BACK RTS CONTROL LINE.
8. DTR INTERACTIONS TEST - VERIFIES THAT CHANGING THE STATE OF THE DTR CONTROL SIGNAL ON ANY LINE DOES NOT AFFECT THE STATE OF ANY STATUS SIGNALS THAT IT IS NOT LOOPED BACK TO.
9. RTS INTERACTIONS TEST - VERIFIES THAT CHANGING THE STATE OF THE RTS CONTROL SIGNAL ON ANY LINE DOES NOT AFFECT THE STATE OF ANY STATUS SIGNALS THAT IT IS NOT LOOPED BACK TO.
10. REPORT BMP CODES TEST THIS PSEUDO TEST REPORTS THE FIRST 32 CHARACTERS WHICH WERE DISCOVERED IN THE FIFO DURING THE EXECUTION OF THE OTHER TESTS. THIS AVOIDS INTERRUPTION OF THE OTHER TESTS BY THESE CODES IF THEY ARE NOT CRITICAL TO THE PERFORMANCE OF THE TESTS.

6.0 EXAMPLE ERROR FREE PASS

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

```
.R CZDHW80
CZDHW80.BIN
DRS
CZDHW-B-0
DHU-11 FUNC TST PART3
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
ACTIVE LINE BIT MAP: (0) 177777 ? <CR>
TYPE OF LOOPBACK (1=INTERNAL, 2=H3029 OR H3277, 3=H325): (0) 2 ?
```

CHANGE SW (L) ? Y

```
REPORT UNIT NUMBER AS EACH UNIT IS TESTED: (L) Y ? <CR>
EXTENDED ERROR REPORTING: (L) N ? Y
NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: (D) 0 ? <CR>
```

TESTING UNIT : 0

TESTING UNIT : 1

```
CZDHW EOP      1
      0 TOTAL ERRS
```

DR>

```

1021                                .LIST SEQ,LOC,BIN,MEB
1022                                .NLIST CND
1030
1031
1032                                .SBTTL  PROGRAM HEADER
1033
1034
1035                                .MCALL  SVC
1036 000000                        SVC                                ; INITIALIZE SUPERVISOR MACROS
1037
1038                                ;*****
1039                                ; IF STRUCTURED MACROS ARE TO BE USED, ADD ".MCALL STRUCT" AND "STRUCT"
1040                                ; TO INITIALIZE THE STRUCTURED MACROS.
1041                                ;*****
1042                                SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
1043                                SVCTST= 1      ; LIST TEST TAGS, SHIFTED RIGHT
1044                                SVCSUB= 1      ; LIST SUBTEST TAGS, SHIFTED RIGHT
1045                                SVCGBL= 1      ; LIST GLOBAL TAGS, SHIFTED RIGHT
1046                                SVCTAG= 1      ; LIST OTHER TAGS, SHIFTED RIGHT
1047
1048                                ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1049                                ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
1050                                ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
1051                                ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1052                                ;*****
1053                                ;*****
1054 000000                        .ENABL  ABS
1055                                ;.ENABL AMA
1056                                "                " 2000
1057
1058 002000                        BGNMOD
1059
1060                                ;**
1061                                ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1062                                ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1063                                ; -
1064
1065 002000                        POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL
1066
1083
1084 002000                        HEADER  CZDHW,B,0,10,0,PRI07
1084 002000
1084 002000                        103
1084 002001                        132
1084 002002                        104
1084 002003                        110
1084 002004                        127
1084 002005                        000
1084 002006                        000
1084 002007                        000
1084 002010
1084 002010                        102
1084 002011
1084 002011                        060
1084 002012
1084 002012 000000
1084 002014

```

```

L$NAME::
        .ASCII /C/
        .ASCII /Z/
        .ASCII /D/
        .ASCII /H/
        .ASCII /W/
        .BYTE  0
        .BYTE  0
        .BYTE  0
L$REV::
        .ASCII /B/
L$DEPO::
        .ASCII '0
L$UNIT::
        .WORD  0
L$TIML::

```

L2

002014 000010
 002016
 002016 017702
 002020
 002020 020076
 002022
 002022 002152
 002024
 002024 002162
 002026
 002026 020400
 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 004060
 002062
 002062 011552
 002064
 002064 000000
 002066
 002066 000000
 002070
 002070 000000
 002072
 002072 012410
 002074
 002074 000000
 002076
 002076 004070
 002100
 002100 104035
 002102
 002102 004010
 002104
 002104 011566

L\$HPCP:: .WORD 10
 L\$SPCP:: .WORD L\$HARD
 L\$HPTP:: .WORD L\$SOFT
 L\$SPTP:: .WORD L\$HW
 L\$LADP:: .WORD L\$SW
 L\$STA:: .WORD L\$LAST
 L\$CO:: .WORD 0
 L\$DTP:: .WORD 0
 L\$APT:: .WORD 0
 L\$DTP:: .WORD 0
 L\$PRI0:: .WORD L\$DISPATCH
 L\$ENVI:: .WORD PRI07
 L\$EXP1:: .WORD 0
 L\$MREV:: .WORD 0
 L\$EF:: .BYTE C\$REVISION
 .BYTE C\$EDIT
 L\$SPC:: .WORD 0
 L\$DEVP:: .WORD 0
 L\$REPP:: .WORD L\$DVTYP
 L\$EXP4:: .WORD L\$RPT
 L\$EXP5:: .WORD 0
 L\$AUT:: .WORD 0
 L\$DUT:: .WORD 0
 L\$LUN:: .WORD L\$DU
 L\$DESP:: .WORD 0
 L\$LOAD:: .WORD L\$DESC
 L\$ETP:: EMT E\$LOAD
 L\$ICP:: .WORD L\$ERRTBL
 .WORD L\$INIT

002106
002106 012372
002110
002110 012370
002112
002112 011560
002114
002114 000000
002116
002116 000000
002120
002120 000000
1085

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

```
1097
1098
1099
1100
1101
1102
1103
1104 002122
      002122 000012
      002124
      002124 012526
      002126 013010
      002130 013544
      002132 014300
      002134 014750
      002136 015420
      002140 016070
      002142 016540
      002144 017170
      002146 017620
1105
```

```
.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 10

                                .WORD 10
L$DISPATCH::
                                .WORD T1
                                .WORD T2
                                .WORD T3
                                .WORD T4
                                .WORD T5
                                .WORD T6
                                .WORD T7
                                .WORD T8
                                .WORD T9
                                .WORD T10
```

1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130

.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.

1131 002150
002150 000003
002152
002152

BGNHW DFPTBL

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

1132
1133 002152 160460
1134 002154 177777
1135 002156 002
1136
1137
1138 002160
002160

.WORD 160460 ;DEFAULT CSR ADDRESS
.WORD 177777 ;DEFAULT ACTIVE LINES BIT MAP
.BYTE 2 ;DEFAULT LOOPBACK TYPE
.EVEN

ENDHW

L10000:

```

1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158 002160
      002160 000002
      002162
      002162
1159
1160 002162 000020
1161 002164 000000
1162
1163 002166
      002166

      .SBTTL SOFTWARE P TABLE
      ;**
      ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
      ; PROGRAM AS OPERATIONAL PARAMETERS. THESE PARAMETERS ARE
      ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
      ; AT RUN TIME.
      ;--
      BGNSW SFPTBL
                                     .WORD L10001-L$SW/2
      L$SW::
      SFPTBL::
      OPTION:: .WORD 20 ;BIT MAP OF PROGRAM CONTROL FLAGS
      NDERPT:: .WORD 0  ;DEFAULT NUMBER OF INDIVIDUAL DATA ERRORS TO RPT.
      ENDSW
                                     L10001:

```

1172
1173
1174
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1227 002166

000020
177777

000000
000002
000002
000004
000006
000006
000010
000012
000014
000016

000020
000030
000100

.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 ;RECTEIVE TIMER REGISTER OFFSET FROM THE CSR ADDRESS
LPRO==4 ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
FSLSO==6 ;FIFOSIZE/STATUS REGISTER OFFSET FROM THE CSR ADDRESS
FDATO==6 ;FIFODATA 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

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

001000
000400
000200
000100

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

```

000040      BITS== 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

```

1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250 002166 000200
1251 002170 000204
1252 002172 177777
1253 002174 000
1254 002175 004
1255 002176 000000
1256
1257
1258
1259
1260 002200
1261 002200 160020
1262 002202 160022
1263 002204 160024
1264 002206 160026
1265
1266 002210 160030
1267 002212 160032
1268 002214 160034
1269 002216 160036
1270
1271
1272
1273
1274 002220 000000
1275 002222 000000
1276 002224 000000
1277 002226 000001
1278 002230 000000
1279 002232 031463
1280 002234 146314
1281 002236 000000
1282 002240 000000
1283 002242 000000
1284 002244 000000
1285 002246 000000
1286 002250 000000
1287 002252 000000
1288 002254 000000
1289
1290
1291
1292 002256 177546
1293 002260 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.
LOPBCK:: .BYTE 0 ;LOOPBACK MODE
BRLEVL:: .BYTE 4 ;INTERRUPT BUS REQUEST LEVEL
UNITN:: .WORD 0 ;UNIT NUMBER.

; DEVICE REGISTER ADDRESS TABLE

DRADRT::
CSRA:: .WORD 160020 ;DMU-11 CSR ADDRESS.
RXTMA:: RBUFA:: .WORD 160022 ;DMU-11 RECIEVE BUFFER/TIMER ADDRESS.
LPRA:: .WORD 160024 ;DMU-11 LINE PARAMETER REGISTER ADDRESS.
FDATA:: FLSA:: .WORD 160026 ;DMU-11 FIFO SIZE/LINE STATUS REGISTER ADDRESS.
;AND FIFO DATA REGISTER ADDRESS.
LNCTRA:: .WORD 160030 ;DMU-11 LINE CONTROL REGISTER ADDRESS.
TXAD1A:: .WORD 160032 ;DMU-11 TRANSMIT BUFFER 1 REGISTER ADDRESS
TXAD2A:: .WORD 160034 ;DMU-11 TRANSMIT BUFFER 2 REGISTER ADDRESS
TXBFCA:: .WORD 160036 ;DMU-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.
EXOERR:: .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.
TP4VEC:: .WORD 0 ;STORAGE FOR THE NORM'L 004 TRAP VECTOR.
TP4FLG:: .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.

1294 002262 000100
 1295 002264 000074
 1296 002266 000000
 1297 002270 000000
 1298 002272 000170
 1299 002274 000170
 1300 002276 000021
 1301 002300 000062
 1302
 1303
 1304
 1305
 1306 002302 177572
 1307 002304 000000
 1308 002306 000000
 1309
 1310
 1311
 1312
 1313 002310 000001
 1314 002312 000002
 1315 002314 000004
 1316 002316 000010
 1317 002320 000020
 1318 002322 000040
 1319 002324 000100
 1320 002326 000200
 1321 002330 000400
 1322 002332 001000
 1323 002334 002000
 1324 002336 004000
 1325 002340 010000
 1326 002342 020000
 1327 002344 040000
 1328 002346 100000
 1329
 1330
 1331
 1332
 1333 002350 000000
 1334 002352
 1335 002552
 1336
 1337
 1338
 1339
 1340 002552 000000
 1341 002554
 1342
 1343
 1344
 1345
 1346 002610
 1347 002610 000000
 1348 002612 000000
 1349 002614 000000
 1350 002616 000000

```

CLKVEC:: .WORD 100 ;INTERRUPT VECTOR ADDRESS OF THE LTC.
CLKHRZ:: .WORD 60. ;INTERRUPT FREQUENCY OF THE LTC.
TIMER1:: .WORD 0 ;HARDWARE CLOCK COUNTER #1.
TIMER2:: .WORD 0 ;HARDWARE CLOCK COUNTER #2.
TIMER3:: .WORD 120. ;HARDWARE BREAK COUNTER LOCATION.
BCOUNT:: .WORD 120. ;BREAK COUNT VALUE IN CLOCK TICKS.
MSTICK:: .WORD 17. ;NUMBER OF MILLI-SECONDS PER LTC TICK.
MSLCNT:: .WORD 62 ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.

;*****
; MEMORY MANAGEMENT VARIABLES AND FLAGS.
;*****
MMSRO:: .WORD 177572 ;ADDRESS OF MEM MGT STATUS REGISTER #0.
MMPRES:: .WORD 0 ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
MMENAB:: .WORD 0 ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).

;*****
; TABLE OF WORDS WITH CORRESPONDING BIT SET FOR GENERATION OF BIT MAPS.
;*****
BITTBL:: .WORD 1 ;BIT 0 SET.
          .WORD 2 ;BIT 1 SET.
          .WORD 4 ;BIT 2 SET.
          .WORD 10 ;BIT 3 SET.
          .WORD 20 ;BIT 4 SET.
          .WORD 40 ;BIT 5 SET.
          .WORD 100 ;BIT 6 SET.
          .WORD 200 ;BIT 7 SET.
          .WORD 400 ;BIT 8 SET.
          .WORD 1000 ;BIT 9 SET.
          .WORD 2000 ;BIT 10 SET.
          .WORD 4000 ;BIT 11 SET.
          .WORD 10000 ;BIT 12 SET.
          .WORD 20000 ;BIT 13 SET.
          .WORD 40000 ;BIT 14 SET.
          .WORD 100000 ;BIT 15 SET.

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

;*****
; STORAGE AREA FOR ERROR SUMMARY TABLE AND FLAGS.
;*****
ERSMRF:: .WORD 0 ;ERROR SUMMARY FLAGS.
ERCNTB:: .BLKW 16 ;TABLE OF ERROR COUNTS.

;*****
; STORAGE AREA FOR THE CONTENTS OF THE DUT STAT REGISTER STATES.
;*****
STSTB:: ;BASE OF DUT STAT STORAGE TABLE.
        .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 0.
        .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 1.
        .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 2.
        .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 3.

```

```

1351 002620 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 4.
1352 002622 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 5.
1353 002624 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 6.
1354 002626 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 7.
1355 002630 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 8.
1356 002632 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 9.
1357 002634 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 10.
1358 002636 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 11.
1359 002640 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 12.
1360 002642 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 13.
1361 002644 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 14.
1362 002646 000000 .WORD 0 ;STORAGE FOR STAT REGISTER FOR LINE 15.
1363 002650 STSTE:: ;END OF DUT STAT STORAGE TABLE.
1364
1365 ;*****
1366 ; GENERAL TABLE AND BUFFER AREA--513 WORDS.
1367 ;*****
1368 002650 BUFBAS:: ;BASE OF MEMORY BUFFER.
1369 002650 ERLTBL:: .BLKW 128. ;FIRST HALF OF GENERAL TABLE OR BUFFER.
1370 003250 BUFMID:: .BLKW 64. ;SECOND HALF OF GENERAL TABLE OR BUFFER.
1371 003450 BUF3QT:: .BLKW 64. ;LAST QUARTER OF THE BUFFER AREA.
1372 003650 BUFEND:: ;END OF GENERAL PURPOSE MEMORY BUFFER.
1373 003650 ENDETBL:: .BLKW 16. ;BUFFER OVERFLOW SPACE.
1374
1375 ;*****
1376 ;* TABLE FOR STORAGE OF RX/TX LINE NUMBER ASSOCIATIONS.
1377 ;* THE ASSOCIATIONS ARE STORED AS LINE NUMBER TIMES 2 FOR USE AS OFFSETS
1378 ;* WHEN ACCESSING A TABLE OF WORDS.
1379 ;* NOTE: DO NOT WRITE A NON-ZERO VALUE INTO THE UPPER BYTE OF ANY ENTRY.
1380 ;*****
1381 003710 TXRXLB:: ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1382 003710 .WORD 0 ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
1383 003712 .WORD 2. ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
1384 003714 .WORD 4. ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
1385 003716 .WORD 6. ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
1386 003720 .WORD 8. ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
1387 003722 .WORD 10. ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
1388 003724 .WORD 12. ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
1389 003726 .WORD 14. ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
1390 003730 .WORD 16. ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
1391 003732 .WORD 18. ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
1392 003734 .WORD 20. ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
1393 003736 .WORD 22. ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
1394 003740 .WORD 24. ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
1395 003742 .WORD 26. ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
1396 003744 .WORD 28. ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
1397 003746 .WORD 30. ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
1398 003750 TXRXLE:: ;END OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1399 .EVEN ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1400
1401 ;*****
1402 ;* TABLE FOR STORAGE OF RX/TX LINE NUMBER ASSOCIATIONS.
1403 ;* THE ASSOCIATIONS ARE STORED AS LINE NUMBERS WHICH CAN BE USED AS SUCH OR
1404 ;* AS OFFSETS WHEN ACCESSING A TABLE OF BYTES.
1405 ;*****
1406 003750 TXRLNB:: ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1407 003750 .BYTE 0 ;TX/RX LINE FOR RX/TX LINE 0.

```

1408 003751 001
 1409 003752 002
 1410 003753 003
 1411 003754 004
 1412 003755 005
 1413 003756 006
 1414 003757 007
 1415 003760 010
 1416 003761 011
 1417 003762 012
 1418 003763 013
 1419 003764 014
 1420 003765 015
 1421 003766 016
 1422 003767 017
 1423 003770

.BYTE 1.
 .BYTE 2.
 .BYTE 3.
 .BYTE 4.
 .BYTE 5.
 .BYTE 6.
 .BYTE 7.
 .BYTE 8.
 .BYTE 9.
 .BYTE 10.
 .BYTE 11.
 .BYTE 12.
 .BYTE 13.
 .BYTE 14.
 .BYTE 15.

;TX/RX LINE FOR RX/TX LINE 1.
 ;TX/RX LINE FOR RX/TX LINE 2.
 ;TX/RX LINE FOR RX/TX LINE 3.
 ;TX/RX LINE FOR RX/TX LINE 4.
 ;TX/RX LINE FOR RX/TX LINE 5.
 ;TX/RX LINE FOR RX/TX LINE 6.
 ;TX/RX LINE FOR RX/TX LINE 7.
 ;TX/RX LINE FOR RX/TX LINE 8.
 ;TX/RX LINE FOR RX/TX LINE 9.
 ;TX/RX LINE FOR RX/TX LINE 10.
 ;TX/RX LINE FOR RX/TX LINE 11.
 ;TX/RX LINE FOR RX/TX LINE 12.
 ;TX/RX LINE FOR RX/TX LINE 13.
 ;TX/RX LINE FOR RX/TX LINE 14.
 ;TX/RX LINE FOR RX/TX LINE 15.

TXRLNE::

.EVEN

;END OF TX/RX LINE NUMBER ASSOCIATION TABLE.
 ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.

 ;* TABLE OF TX/RX LINE NUMBER ASSOCIATIONS IN STAGGERED LOOPBACK.
 ;* THE ASSOCIATIONS ARE STORED AS LINE NUMBER TIMES 2 FOR USE AS OFFSETS
 ;* WHEN ACCESSING A TABLE OF WORDS.
 ;* THIS IS A TABLE OF DATA FOR READING ONLY. USE TO LOAD THE ABOVE TABLE.
 ;* NOTE: MUST CONVERT FROM BYTES TO WORDS WHEN LOADING ABOVE TABLE.
 ;*****

STGTRB::

.BYTE 4.
 .BYTE 6.
 .BYTE 0
 .BYTE 2.
 .BYTE 12.
 .BYTE 14.
 .BYTE 8.
 .BYTE 10.
 .BYTE 20.
 .BYTE 22.
 .BYTE 16.
 .BYTE 18.
 .BYTE 28.
 .BYTE 30.
 .BYTE 24.
 .BYTE 26.
 .EVEN

;BASE OF STAGGERED TX/RX LINE NUMBER TABLE.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
 ;TX/RX LINE OFFSET FOR RX/TX LINE 15.

;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.

1433 003770
 1434 003770 004
 1435 003771 006
 1436 003772 000
 1437 003773 002
 1438 003774 014
 1439 003775 016
 1440 003776 010
 1441 003777 012
 1442 004000 024
 1443 004001 026
 1444 004002 020
 1445 004003 022
 1446 004004 034
 1447 004005 036
 1448 004006 030
 1449 004007 032
 1450
 1451
 1464 004010
 004010
 004010 000000
 004012 000000
 004014 000000
 004016 000000

ERRTBL

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

L\$ERRTBL::

1465
 1466

.EVEN

```

1468 .SBTTL GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.
1469 :*****
1470 :* THERE ARE 4 ROUTINES AND MACRO DEFINITIONS USED FOR THE HANDLING OF
1471 :* GPR VALUES DURING SUBROUTINE CALLS WITHIN THIS PROGRAM. THE FOUR
1472 :* ROUTINES/MACRO CALLS HAVE THE FOLLOWING NAMES:
1473 :*
1474 :* SAVE - MACRO DEFINITION USED AT THE BEGINNING OF A SUBROUTINE TO
1475 :* SAVE THE GPR CONTENTS FOR LATER RESTORATION.
1476 :* PASS - MACRO DEFINITION USED AT THE END OF A SUBROUTINE TO RESTORE
1477 :* THE PREVIOUSLY SAVED GPR CONTENTS AND TO LEAVE THE CONTENTS
1478 :* OF THE SPECIFIED GPR(S) INTACT (NOT RESTORED).
1479 :* PREG05 - SUBROUTINE WHICH IS CALLED FROM THE SAVE AND PASS MACRO
1480 :* EXPANSIONS WHICH ACTUALLY PERFORMS THE ACTIONS ON THE GPRS.
1481 :*
1482 :* DURING A SUBROUTINE WHICH USES THESE GPR SAVE ROUTINES THE VALUES
1483 :* OF THE GPRS ARE STORED ON THE STACK IN THE FOLLOWING STACK FRAME:
1484 :*
1485 :* SP -> RET PC INTO PREG05 ROUTINE.
1486 :* SP+2 -> GPR R0 CONTENTS.
1487 :* SP+4 -> GPR R1 CONTENTS.
1488 :* SP+6 -> GPR R2 CONTENTS.
1489 :* SP+8 -> GPR R3 CONTENTS.
1490 :* SP+10 -> GPR R4 CONTENTS.
1491 :* SP+12 -> GPR R5 CONTENTS.
1492 :* SP+14 -> RET PC INTO CALLER OF SUB'TNE WHICH CALLED PREG05.
1493 :*
1494 :* EACH LEVEL OF SUB'TNE CALLING USES 8 WORDS OF STACK OVERHEAD.
1495 :* THE SAVE AND PASS MACROS CAN ALSO BE USED IN "STRAIGHT LINE CODE"
1496 :* TO SAVE AND RESTORE THE GPR VALUES. IN ANY CASE, AFTER THE
1497 :* ISSUING OF A PASS CALL THE GPRS WILL BE RESTORED TO THE VALUES
1498 :* THEY HAD PRIOR TO THE LAST SAVE CALL (EXCEPT FOR THE EXCEPTED,
1499 :* OR PASSED INTACT, GPRS SPECIFIED AS PARAMETERS TO THE PASS CALL)
1500 :* AND THE SP WILL ALSO BE RESTORED TO ITS CONDITION BEFORE THE LAST
1501 :* SAVE CALL. THE PROGRAMMER MUST BE SURE THAT THE SP HAS THE SAME
1502 :* VALUE WHEN THE PASS MACRO IS CALLED AS IT HAD IMMEDIATELY AFTER
1503 :* THE SAVE MACRO WAS CALLED.
1504 :*****

```

```
1506
1507
1508
1509
1510
1511
1512
1513      000036
1514      000016
1515      000014
1516      000012
1517      000010
1518      000006
1519      000004
1520      000002

.SBTTL GPR FRAME ACCESS EQUATES
;***
;EQUATES THAT ALLOW ACCESS TO THE STACK FRAME.  THESE ARE THE
;OFFSETS INTO THE STACK FOR REGISTERS SAVED DURING THE PREGOS
;ROUTINE.
; --
LPCSLT==      36      ;OFFSET FOR LAST RETURN PC.
PCSLT==      16      ;OFFSET FOR RETURN PC.
R5SLT==      14      ;OFFSET FOR R5.
R4SLT==      12      ;OFFSET FOR R4.
R3SLT==      10      ;OFFSET FOR R3.
R2SLT==       6      ;OFFSET FOR R2.
R1SLT==       4      ;OFFSET FOR R1.
R0SLT==       2      ;OFFSET FOR R0.
```

1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545

```
.SBTTL GLOBAL MACRO DEFINITION                                SAVE
;*****
;* THIS MACRO IS USED AT THE BEGINNING OF A SUBROUTINE TO SAVE THE
;* CONTENTS OF THE GPRS R0 THRU R5.
;*
;* INPUTS:      SP    UNCHANGED SINCE SUBROUTINE WAS ENTERED
;*              R5SLOT - OFFSET TO STACK SLOT FOR R5 (EQUATED TO 14 OCTAL)
;*
;* OUTPUTS:     GPR SAVE AREA ON THE STACK IS LOADED WITH THE CONTENTS OF GPRS
;*              TOP OF STACK   LOADED WITH THE RETURN ADDRESS INTO PREG05
;*
;* CALLING SEQUENCE:  SAVE
;*
;* COMMENTS:      NO ARGUMENTS ARE ALLOWED.
;*                THE PASS MACRO SHOULD BE CALLED TO RESTORE THE GPR VALUES.
;*
;* SUBORDINATE ROUTINES CALLED: PREG05.
;*****
               .MACRO  SAVE
               .LIST
               JSR     R5,PREG05           ;CALL REGISTER SAVE SUBRT.
               .NLIST
               .ENDM   SAVE
```

1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594

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

```
1596 .SBTTL GLOBAL SUBROUTINE PREG05 -
1597 ;*****
1598 ;* PRESERVE REGISTERS R0 THROUGH R5 FOR SUBROUTINE CALLS.
1599 ;*
1600 ;* INPUTS: THE RETURN ADDRESS BACK INTO THE CALLING ROUTINE MUST BE IN
1601 ;* GPR R5. (I.E. MACROS USE "JSR R5,PREG05".)
1602 ;*
1603 ;* OUTPUTS: REGISTERS R0 THROUGH R5 ARE SAVED ON THE STACK.
1604 ;*
1605 ;* CALLING SEQUENCE: SAVE ;MACRO EXPANSION CALLS PREG05.
1606 ;* [SUBROUTINE CODE]...
1607 ;* PASS ;MACRO EXPANSION RECALLS PREG05.
1608 ;*
1609 ;* COMMENTS: THIS ROUTINE IS RE ENTRANT.
1610 ;*
1611 ;* PARAMETERS MAY BE PASSED OUT OF A SUBROUTINE BY MODIFYING THE
1612 ;* REGISTER SAVE AREA ON THE STACK. USE THE PASS GPRN MACRO
1613 ;* TO RETURN GPR VALUES INTACT.
1614 ;* USE THE RNSLOT OFFSETS FROM THE SP TO PASS OTHER PARAMETERS.
1615 ;* [EXAMPLE: MOV VALUE,R0SLOT(SP) ]
1616 ;* MAKE SURE THE SP IS AT ITS ORIGINAL VALUE WHEN YOU DO THIS.
1617 ;*
1618 ;* SUBORDINATE ROUTINES CALLED: NONE.
1619 ;*****
1620
1621 004020 PREG05: ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
1622 004020 010446 MOV R4,-(SP) ;SAVE R4
1623 004022 010346 MOV R3,-(SP) ;SAVE R3
1624 004024 010246 MOV R2,-(SP) ;SAVE R2
1625 004026 010146 MOV R1,-(SP) ;SAVE R1
1626 004030 010046 MOV R0,-(SP) ;SAVE R0
1627 004032 010546 MOV R5,-(SP) ;PUSH RETURN PC ON TOP OF STACK
1628 004034 016605 000014 MOV R5SLOT(SP),R5 ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS
1629
1630 004040 004736 JSR PC,@(SP)+ ;CALL THE SUBROUTINE AT THE RETURN ADDRESS
1631 ;FROM THE PREG05 CALL, PUTTING THE PRESENT
1632 ;PC ON THE STACK AS A RETURN ADDRESS INTO
1633 ;THIS (PREG05) ROUTINE.
1634
1635 ;***
1636 ;THE FOLLOWING CODE IS EXECUTED WHEN THE CALLING ROUTINE DOES A
1637 ;"RETURN" [JSR PC,@(SP)+] USING THE PC DEPOSITED ON THE STACK ABOVE.
1638 ;---
1639
1640 004042 012605 PREGRT: MOV (SP)+,R5 ;PUT RETURN PC IN R5.
1641 004044 012600 MOV (SP)+,R0 ;RESTORE R0.
1642 004046 012601 MOV (SP)+,R1 ;RESTORE R1.
1643 004050 012602 MOV (SP)+,R2 ;RESTORE R2.
1644 004052 012603 MOV (SP)+,R3 ;RESTORE R3.
1645 004054 012604 MOV (SP)+,R4 ;RESTORE R4.
1646
1647 004056 000205 RTS R5 ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05.
1648 ;RESTORING R5 IN THE PROCESS.
```

```

1650          .SBTTL  GLOBAL TEXT SECTION
1658
1659
1660
1661          ;
1662          ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1663          ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1664          ; MORE THAN ONE TEST.
1665          ;
1666          ;
1667          ; NAMES OF DEVICES SUPPORTED BY PROGRAM
1668          ;
1669          004060          DEVTYP  <DMU 11>
          004060
          004060          104      110      125          L$DVTYP::
          004063          055      061      061          .ASCIIZ  /DMU 11/
          004066          000
                                     .EVEN
1670
1676          ; TEST DESCRIPTION
1677          ;
1678          ;
1679          004070          DESCRIPT  <DMU 11 FUNC TST PART3>
          004070
          004070          104      110      125          L$DESC::
T3/          004073          055      061      061          .ASCIIZ  /DMU-11 FUNC TST PAR
          004076          040      106      125
          004101          116      103      040
          004104          124      123      124
          004107          040      120      101
          004112          122      124      063
          004115          000
1680          .EVEN
1681
1688          .EVEN

```

```

1697
1698      .NLIST BIN
1699
1700
1701      ; ***** FORMAT STATEMENTS USED IN PRINT CALLS *****
1702
1703
1704 004116 EF0503:: .ASCIZ /#T#N/
1705 004123 EF1601:: .ASCIZ /#A #T#A, TEST ABORTED #N/
1706 004155 EF7801:: .ASCIZ /#T#A ON LINE #D2#A DECIMAL.#N/
1707 004213 EF8401:: .ASCIZ /#A #T#A FOR LINE #D2#A(D) AFFECTS OTHER MODEM SIGNALS.#N/
1708 004305 EF8402:: .ASCIZ /#A          CHANGING #T#A FOR LINE #D2#A(D) AFFECTED /
1709 004371          .ASCIZ /#T#A FOR LINE #D2#A(D).#N/
1710 004423 EF9301:: .ASCIZ /#A #T#D2#A(D), BMP CODE REPORTED :#03#A(0)#N/
1711 004501 EF9302:: .ASCIZ /#A OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)#N/
1712 004601 MFUNIT:: .ASCIZ /#N#A TESTING UNIT :#D4#N/
1713      .EVEN
1714      .LIST BIN

```

```

1723
1724      .NLIST BIN
1725
1726
1727      ;***** GLOBAL ERROR MESSAGES *****
1728
1729 004632 EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
1730 004670 EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
1731 004753 EM7801:: .ASCIZ /MODEM CONTROL DTR BIT TEST FAILED/
1732 005015 EM7802:: .ASCIZ / DTR BIT FAULTY/
1733 005035 EM7901:: .ASCIZ /MODEM CONTROL RTS BIT TEST FAILED/
1734 005077 EM7902:: .ASCIZ / RTS BIT FAULTY/
1735 005117 EM8001:: .ASCIZ /DSR MODEM STATUS SIGNAL TEST FAILED/
1736 005163 EM8002:: .ASCIZ / DSR MODEM STATUS SIGNAL DEFECTIVE/
1737 005227 EM8101:: .ASCIZ /RI MODEM STATUS SIGNAL TEST FAILED/
1738 005272 EM8102:: .ASCIZ / RI MODEM STATUS SIGNAL DEFECTIVE/
1739 005335 EM8201:: .ASCIZ /CTS MODEM STATUS SIGNAL TEST FAILED/
1740 005401 EM8202:: .ASCIZ / CTS MODEM STATUS SIGNAL DEFECTIVE/
1741 005445 EM8301:: .ASCIZ /DCD MODEM STATUS SIGNAL TEST FAILED/
1742 005511 EM8302:: .ASCIZ / DCD MODEM STATUS SIGNAL DEFECTIVE/
1743 005555 EM8401:: .ASCIZ /DTR MODEM CONTROL SIGNAL INTERACTIONS TEST FAILED/
1744 005637 EM8402:: .ASCIZ /DTR/
1745 005643 EM8403:: .ASCIZ /DSR/
1746 005647 EM8404:: .ASCIZ /RI/
1747 005652 EM8405:: .ASCIZ /DCD/
1748 005656 EM8406:: .ASCIZ /CTS/
1749 005662 EM8501:: .ASCIZ /RTS MODEM CONTROL SIGNAL INTERACTIONS TEST FAILED/
1750 005744 EM8502:: .ASCIZ /RTS/
1751 005750 EM9301:: .ASCIZ /BMP CODES WERE REPORTED DURING THIS DIAGNOSTIC/
1752 006027 EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
1753 006057 EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
1754 006124 EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
1755      .EVEN
1756      .LIST BIN

```

1765
1766
1767
1768
1769
1770
1771
1772
1773

.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.

```
1775 .SBTTL GLOBAL ERROR REPORTING ROUTINE ER0101 -
1776 ;*****
1777 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1778 ;* INFORMATION IF AN ERROR IS DETECTED IN TEST 1 (REGISTER ADDRESS
1779 ;* ACCESS TEST). IF THE "EXTENDED ERROR INFO" OPTION HAS BEEN SELECTED
1780 ;* THEN THIS SUBROUTINE WILL REPORT THE TYPE OF ACCESS (READ OR WRITE OR
1781 ;* BOTH) WHICH CAUSED A BUS TIME-OUT TRAP (004 TRAP). A MESSAGE INDICATING
1782 ;* THAT THE DHU MAY BE AT THE WRONG UNIBUS ADDRESS IS ALSO PRINTED.
1783 ;*
1784 ;* INPUTS: R5 - ERROR FLAG WORD.
1785 ;* IF BIT 0 IS SET, A READ ERROR OCCURED.
1786 ;* IF BIT 1 IS SET, A WRITE ERROR OCCURED.
1787 ;*
1788 ;* OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE.
1789 ;*
1790 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER0101" AS THE MESSAGE POINTER
1791 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
1792 ;*
1793 ;* COMMENTS:
1794 ;*
1795 ;* SUBORDINATE ROUTINES USED: NONE.
1796 ;*****
1797
1798 006200 BGNMSG ER0101
1799 006200 ER0101::
006200 004567 175614 SAVE JSR ;SAVE THE GPR CONTENTS.
1800 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
1801 006204 012700 000100 MOV #BIT06,R0 ;SET-UP THE BIT MAP FOR 'REPORT EXT'D ERROR INFO'
1802 006210 046700 173746 BIC OPTION,R0 ;TRY AND CLEAR THE FLAG.
1803 006214 001036 BNE 6$ ;EXIT IF OPTION NOT SELECTED.
1804 ;*
1805 ;* REPORT EXTENDED ERROR INFORMATION
1806 ;*
1807 ;*
1808 006216 032705 000001 BIT #BIT0,R5 ;TEST FOR READ ERROR.
1809 006222 001410 BEQ 2$ ;SKIP READ ERROR MSG IF NO READ ERROR.
1810 006224 012746 006316 PRINTB #MSG1 ;PRINT READ ERROR MESSAGE.
1811 006230 012746 000001 MOV #MSG1,-(SP)
1812 006234 010600 MOV #1,-(SP)
1813 006236 104414 MOV SP,R0
1814 006240 062706 000004 TRAP C$PNTB
1811 006244 032705 000002 2$: BIT #BIT1,R5 ;TEST FOR WRITE ERROR.
1812 006250 001410 BEQ 4$ ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
1813 006252 012746 006374 PRINTB #MSG2 ;PRINT WRITE ERROR MESSAGE.
1814 006256 012746 000001 MOV #MSG2,-(SP)
1815 006262 010600 MOV #1,-(SP)
1816 006264 104414 MOV SP,R0
1817 006266 062706 000004 TRAP C$PNTB
1818 006272 012746 006453 4$: PRINTX #MSG3 ;SUGGEST THAT DHU MAY BE AT WRONG ADDRESS.
1819 006276 012746 000001 MOV #MSG3,-(SP)
1820 006302 010600 MOV #1,-(SP)
1821 006304 104415 MOV SP,R0
1822 006306 062706 000004 TRAP C$PNTX
1823 ADD #4,SP
```

```
1815 006312          6$: PASS          ;RESTORE THE GPR CONTENTS.
      006312 004736          JSR      PC,0(SP).      ;RETURN TO PREGOS SUBRT.
1816 006314          ENDMSG
      006314          L10002:
      006314 104423          TRAP      C$MSG
1817
1818 006316          045      101      102 MSG1:: .ASCIZ /#ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.#N/
      006321          125      123      040
      006324          124      111      115
      006327          105      055      117
      006332          125      124      040
      006335          124      122      101
      006340          120      040      103
      006343          101      125      123
      006346          105      104      040
      006351          102      131      040
      006354          122      105      101
      006357          104      040      101
      006362          124      124      105
      006365          115      120      124
      006370          056      045      116
      006373          000
1819 006374          045      101      102 MSG2:: .ASCIZ /#ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.#N/
      006377          125      123      040
      006402          124      111      115
      006405          105      055      117
      006410          125      124      040
      006413          124      122      101
      006416          120      040      103
      006421          101      125      123
      006424          105      104      040
      006427          102      131      040
      006432          127      122      111
      006435          124      105      040
      006440          101      124      124
      006443          105      115      120
      006446          124      056      045
      006451          116      000
1820 006453          045      101      104 MSG3:: .ASCIZ /#ADHU MAY BE AT THE WRONG UNIBUS ADDRESS.#N#N/
      006456          110      125      040
      006461          115      101      131
      006464          040      102      105
      006467          040      101      124
      006472          040      124      110
      006475          105      040      127
      006500          122      117      116
      006503          107      040      125
      006506          116      111      102
      006511          125      123      040
      006514          101      104      104
      006517          122      105      123
      006522          123      056      045
      006525          116      045      116
      006530          000
1821
1822          .EVEN
```

```
1824 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0503
1825 ;*****
1826 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
1827 ;* MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER, PROVIDED
1828 ;* EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
1829 ;*
1830 ;* INPUTS: R1 ADDRESS OF THE MESSAGE TO PRINT.
1831 ;*
1832 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
1833 ;*
1834 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE MESSAGE IN R1.
1835 ;* INCLUDE THE LABEL "ER0503" AS THE MESSAGE POINTER
1836 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1837 ;*
1838 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
1839 ;*
1840 ;* SUBORDINATE ROUTINES USED: NONE.
1841 ;*****
1842
1843 006532 BGNMSG ER0503
1844 006532
1845 006532 012700 000100
1846 006536 046700 173420
1847 006542 001011
1848
1849
1850 006544 PRINTB #EF050? R1 ;PRINT THE MESSAGE.
1851 006544 010146
1852 006546 012746 004116
1853 006552 012746 000002
1854 006556 010600
1855 006560 104414
1856 006562 062706 000006
1857
1858 006566
1859 006566
1860 006566 104423
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
```

```
ER0503::
    MOV    #BIT06,R0      ;TRY TO CLEAR THE
    BIC    OPTION,R0      ;EXT'D ERROR REPORTING FLAG
    BNE    2$             ;EXIT IF FLAG NOT SET.

    PRINTB #EF050? R1     ;PRINT THE MESSAGE.

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

2$:      ENDMSG

L10003:  TRAP    C$MSG
```

```
1854 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER1603
1855 ;*****
1856 ;* THIS ERROR REPORTING ROUTINE IS USED TO PRINT OUT A BASIC ERROR
1857 ;* MESSAGE, ALONG WITH A MESSAGE INFORMING THE OPERATOR WHICH TEST IS
1858 ;* ABOUT TO BE ABORTED, PROVIDED EXTENDED ERROR INFORMATION HAS BEEN
1859 ;* REQUESTED. OTHERWISE ONLY A "TEST FAILURE" MESSAGE WILL BE PRINTED
1860 ;*
1861 ;* INPUTS: R1 CONTAINS THE ADDRESS OF THE MESSAGE TO BE PRINTED.
1862 ;* ERRMSG - CONTAINS THE ADDRESS OF THE MESSAGE THAT INDICATES
1863 ;* THE TEST THAT IS BEING PERFORMED, EG DMA, BREAK ETC.
1864 ;*
1865 ;* OUTPUTS. MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
1866 ;* "TESTNAME TEST ABORTED"
1867 ;*
1868 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER1603" AS THE MESSAGE POINTER
1869 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
1870 ;*
1871 ;* COMMENTS:
1872 ;*
1873 ;*
1874 ;* SUBORDINATE ROUTINES CALLED: NONE.
1875 ;*****
1876 006570 BGNMSG ER1603
1877 006570
1878 006570 004567 175224
1879 006574 012700 000100
1880 006600 046700 173356
1881 006604 001024
1882
1883
1884 006606
1885 006606 010146
1886 006610 012746 004116
1887 006614 012746 000002
1888 006620 010600
1889 006622 104414
1890 006624 062706 000006
1891 006630 016702 175160
1892 006634
1893 006634 010246
1894 006636 012746 004123
1895 006642 012746 000002
1896 006646 010600
1897 006650 104414
1898 006652 062706 000006
1899
1900 006656
1901 006656 004736
1902 006660
1903 006660 104423
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
```

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

MOV  %BIT06,R0                      ;TRY TO CLEAR THE
BIC  OPTION,R0                     ;EXT'D ERROR REPORTING FLAG
BNE  2$                             ;EXIT IF FLAG NOT SET.

PRINTB #EF0503,R1                  ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
MOV  R1,-(SP)
MOV  #EF0503,-(SP)
MOV  #2,-(SP)
MOV  SP,R0
TRAP C$PNTB
ADD  #6,SP

MOV  ERRMSG,R2                      ;GET THE "TEST MESSAGE".
PRINTB #EF1601,R2                  ;PRINT "TEST ABORTED" MESSAGE.
MOV  R2,-(SP)
MOV  #EF1601,-(SP)
MOV  #2,-(SP)
MOV  SP,R0
TRAP C$PNTB
ADD  #6,SP

2$: PASS                            ;RESTORE THE CONTENTS OF THE GPRS.
JSR PC,@(SP)                       ;RETURN TO PREG05 SUBRT.

ENDMSG

L10004:
TRAP C$MSG
```

```
1892 .SBTTL GLOBAL ERROR REPORTING ROUTINE ER7801
1893 ;*****
1894 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
1895 ;* MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER. A LINE NUMBER
1896 ;* IS INCLUDED AT THE END OF THE MESSAGE. THE MESSAGE IS PRINTED ONLY IF
1897 ;* EXTENDED ERROR REPORTING IS REQUESTED.
1898 ;*
1899 ;* INPUTS: R1 ADDRESS OF THE MESSAGE TO PRINT.
1900 ;* R3 NUMBER OF LINE ON WHICH ERROR OCCURRED.
1901 ;*
1902 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
1903 ;*
1904 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE MESSAGE IN R1.
1905 ;* LOAD THE LINE NUMBER INTO R3.
1906 ;* INCLUDE THE LABEL "ER7801" AS THE MESSAGE POINTER
1907 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1908 ;*
1909 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
1910 ;*
1911 ;* SUBORDINATE ROUTINES USED: NONE.
1912 ;*****
1913
1914 006662 BGNMSG ER7801
1915 006662 ER7801::
1916
1917 ;*
1918 ;* EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
1919 006662 032767 000100 173272 ;*
1920 006670 001412 BIT #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
1921 ;* BEQ 2$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
1922 ;* ;DURING THE SOFTWARE QUESTIONS.
1923 006672 PRINTB #EF7801,R1,R3 ;PRINT THE MESSAGE.
1924 006672 010346 MOV R3,-(SP)
1925 006674 010146 MOV R1,-(SP)
006676 012746 004155 MOV #EF7801,-(SP)
006702 012746 000003 MOV #3,-(SP)
006706 010600 MOV SP,R0
006710 104414 TRAP C$PNTB
006712 062706 000010 ADD #10,SP
1924
1925 006716 2$: ENDMSG
006716
006716 104423 L10005: TRAP C$MSG
```

```
1927 .SBTTL GLOBAL ERROR REPORTING ROUTINE ER8401
1928 ;*****
1929 ;* THIS ERROR REPORTING SUBROUTINE IS INTENDED TO REPORT INTERACTIONS
1930 ;* WHICH HAVE BEEN FOUND BETWEEN A MODEM SIGNAL AND OTHER MODEM SIGNALS.
1931 ;* IT ANALYZES THE MODEM STATUS WHICH IS STORED IN THE STAT STORAGE AREA
1932 ;* AND REPORTS ANY DISCREPANCIES WHICH ARE FOUND BETWEEN THIS STORED DATA
1933 ;* AND THE PRESENT STATE OF THE STAT REGISTERS. SPECIFIED BITS ON THE
1934 ;* LINE ASSOCIATED WITH THE SPECIFIED LINE ARE IGNORED.
1935 ;*
1936 ;* INPUTS: R1 - ADDRESS OF SIGNAL NAME MESSAGE.
1937 ;* R2 - BIT MAP OF BITS TO IGNORE ON SPECIFIED LINE.
1938 ;* R3 - NUMBER OF SPECIFIED LINE.
1939 ;* CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
1940 ;* NUMLNS - EQUATED TO THE NUMBER OF LINES ON THE DUT.
1941 ;* FLSA - CONTAINS THE ADDRESS OF THE DUT STAT REGISTER.
1942 ;* STSTB - LABEL AT BASE OF STAT STORAGE TABLE.
1943 ;* TXRLNB - LABEL AT BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1944 ;*
1945 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
1946 ;*
1947 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER8401" AS THE MESSAGE POINTER
1948 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1949 ;*
1950 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
1951 ;*
1952 ;* SUBORDINATE ROUTINES USED: NONE.
1953 ;*****
1954
1955 006720 BGNMSG ER8401
1956 006720
1957 006720 004567 175074
1958
1959
1960 ;*
1961 006724 032767 000100 173230 ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
1962 006732 001517
1963
1964
1965 006734
006734 010346
006736 010146
006740 012746 004213
006744 012746 000003
006750 010600
006752 104414
006754 062706 000010
1966
1967 006760 010167 000204
1968 006764 005001
1969 006766 012704 002610
1970 006772 010177 173202
1971 006776 017700 173204
1972 007002 011405
1973 007004 040005
1974 007006 042400

;*
;* BIT 0BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
;* BEQ 60$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
;* DURING THE SOFTWARE QUESTIONS.

PRINTB 0EF8401,R1,R3 ;PRINT THE BASIC MESSAGE.

MOV R3,-(SP)
MOV R1,-(SP)
MOV 0EF8401,-(SP)
MOV 03,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD 010,SP

MOV R1,44$ ;SAVE THE ADDRESS OF THE SIGNAL NAME MESSAGE.
CLR R1 ;CLEAR THE LINE COUNTER.
MOV 0STSTB,R4 ;SET UP STAT STORAGE POINTER TO BASE OF TABLE.
MOV R1,0CSRA ;SET UP THE CSR IND.ADR.REG FIELD.
MOV 0FLSA,R0 ;GET THE CONTENTS OF THIS LINE'S STAT REGISTER.
MOV (R4),R5 ;GET THE PREVIOUS CONTENTS FROM STORAGE.
BIC R0,R5
BIC (R4)+,R0
```

```
1975 007010 050005      BIS      R0,R5      ;XOR PRESENT AND STORED STAT VALUES.
1976 007012 012700 043777  MOV      #43777,R0      ;PREPARE TO MASK OUT UNUSED BITS.
1977 007016 120163 003750  CMPB     R1,TXRLNB(R3)      ;IS THIS LINE ASSOCIATED WITH SPECIFIED LINE?
1978 007022 001002      BNE      4$      ;DON'T MASK OUT SPECIFIED BITS IF IT IS NOT.
1979 007024 056600 000006      BIS      R2,SLOT(SP),R0      ;MASK OUT SPECIFIED BITS.
1980 007030 040005      BIC      R0,R5      ;GET BIT MAP OF UNDESIRED CHANGES.
1981 007032 032705 100000 4$:      BIT      #BIT15,R5      ;CHECK FOR DSR SIGNAL INTERACTION.
1982 007036 001404      BEQ      6$      ;SKIP PRINTING LINE IF NO DSR INTERACTION.
1983 007040 012702 005643      MOV      #EM8403,R2      ;SELECT DSR ERROR MESSAGE.
1984 007044 004767 000064      JSR      PC,40$      ;PRINT THE LINE OF THE ERROR MESSAGE.
1985 007050 032705 020000 6$:      BIT      #BIT13,R5      ;CHECK FOR RI SIGNAL INTERACTION.
1986 007054 001404      BEQ      8$      ;SKIP PRINTING LINE IF NO RI INTERACTION.
1987 007056 012702 005647      MOV      #EM8404,R2      ;SELECT RI ERROR MESSAGE.
1988 007062 004767 000046      JSR      PC,40$      ;PRINT THE LINE OF THE ERROR MESSAGE.
1989 007066 032705 010000 8$:      BIT      #BIT12,R5      ;CHECK FOR DCD SIGNAL INTERACTION.
1990 007072 001404      BEQ      10$      ;SKIP PRINTING LINE IF NO DCD INTERACTION.
1991 007074 012702 005652      MOV      #EM8405,R2      ;SELECT DCD ERROR MESSAGE.
1992 007100 004767 000030      JSR      PC,40$      ;PRINT THE LINE OF THE ERROR MESSAGE.
1993 007104 032705 004000 10$:     BIT      #BIT11,R5      ;CHECK FOR CTS SIGNAL INTERACTION.
1994 007110 001404      BEQ      12$      ;SKIP PRINTING LINE IF NO CTS INTERACTION.
1995 007112 012702 005656      MOV      #EM8406,R2      ;SELECT CTS ERROR MESSAGE.
1996 007116 004767 000012      JSR      PC,40$      ;PRINT THE LINE OF THE ERROR MESSAGE.
1997
1998 007122 005201 12$:      INC      R1      ;SELECT NEXT LINE.
1999 007124 020127 000020      CMP      R1,#NUMLNS      ;ALL LINES DONE?
2000 007130 002720      BLT      2$      ;LOOP IF NOT ALL LINES DONE.
2001 007132 000417      BR       60$      ;EXIT THIS ROUTINE.
2002
2003      ;*
2004      ; LOCAL ERROR MESSAGE LINE PRINTING ROUTINE.
2005      ;*
2005 007134      40$:      PRINTX   #EF8402,44$,R3,R2,R1
2006 007134 010146
2007 007136 010246
2008 007140 010346
2009 007142 016746 000022
2006 007146 012746 004305
2007 007152 012746 000005
2008 007156 010600
2009 007160 104415
2006 007162 062706 000014
2007 007166 000207
2008 007170 000000
2009 007172 004736
2009 007174
2009 007174 104423

RTS      PC
44$:     .WORD 0
60$:     PASS

ENDMSG

JSR      PC,@(SP),
;LOCAL STORAGE FOR ADDRESS OF SIGNAL NAME.
;RESTORE ALL THE GPRS TO THE PRESERVED VALUES.
;RETURN TO PREG05 SUBRT.

L10006:
TRAP     C$MSG
```

```
2011 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9301
2012 ;*****
2013 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ANY BMP CODES
2014 ;* THAT ARE FOUND IN THE BMP CODE QUEUE, TOGETHER WITH THE THE NUMBER OF
2015 ;* THE TEST THAT WAS EXECUTING AT THE TIME THE BMP CODE WAS LOGGED.
2016 ;* PROVIDED EXTENDED ERROR REPORTING HAS BEEN ENABLED.
2017 ;*
2018 ;* INPUTS: R1 THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
2019 ;* R2 THE ADDRESS OF THE NEXT EMPTY CELL IN THE QUEUE.
2020 ;*
2021 ;* OUTPUTS: THE TEST NUMBER FOLLOWED BY THE BMP CODE ARE PRINTED AT THE
2022 ;* OPERATOR CONSOLE.
2023 ;*
2024 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9301" AS THE MESSAGE POINTER
2025 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2026 ;*
2027 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2028 ;*
2029 ;* SUBORDINATE ROUTINES USED: NONE.
2030 ;*****
2031
2032 007176 BGNMSG ER9301
2033 007176 ER9301::
2034 007176 004567 174616 SAVE JSR R5,PREG05 ;SAVE THE GPRS ON THE STACK.
2035 007202 012700 000100 ;CALL REGISTER SAVE SUBRT.
2036 007206 046700 172750 MOV #BIT06,R0 ;TRY TO CLEAR THE
2037 007212 001064 BIC OPTION,R0 ;EXT'D ERROR REPORTING FLAG
2038 BNE 60$ ;EXIT IF FLAG NOT SET.
2039 007214 PRINTB #EF0503,R1 ;REPORT UNEXPECTED BMP CODES FOUND.
2040 007214 010146 MOV R1,-(SP)
2041 007216 012746 004116 MOV #EF0503,-(SP)
2042 007222 012746 000002 MOV #2,-(SP)
2043 007226 010600 MOV SP,R0
2044 007230 104414 TRAP C:PNTB
2045 007232 062706 000006 ADD #6,SP
2046 007236 012703 002352 MOV #BMPCQB,R3 ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
2047 007242 012705 006027 MOV #EM9302,R5 ;GET THE MESSAGE TO BE REPORTED.
2048 2$: MOV (R3)+,R1 ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
2049 MOV (R3)+,R4 ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
2050 JSR PC,50$ ;GO REPORT THE BMP CODE.
2051 CMP R3,R2 ;CHECK IF ALL CODES HAVE BEEN REPORTED.
2052 BLO 2$ ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
2053 ;*
2054 ;* CHECK IF OVERFLOW HAS OCCURRED.
2055 ;* THE CONDITIONS FOR OVERFLOW ARE: THE POINTER CONTAINS THE ADDRESS OF THE
2056 ;* LAST CELL IN THE QUEUE, AND A BMP CODE HAS ALREADY BEEN WRITTEN INTO THAT
2057 ;* CELL.
2058 ;*
2059 007262 020227 002546 CMP R2,#BMPCQE-4 ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
2060 007266 001036 BNE 60$ ;EXIT IF NOT AT THE LAST LOCATION.
2061 007270 005762 000002 TST 2(R2) ;CHECK FOR A BMP CODE IN THE LAST CELL
2062 007274 001433 BEQ 60$ ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
2063 007276 012301 MOV (R3)+,R1 ;GET THE TEST NUMBER OFF THE QUEUE.
2064 007300 011304 MOV (R3),R4 ;GET THE BMP CODE OFF THE QUEUE.
2065 007302 012705 006057 MOV #EM9303,R5 ;SELECT THE MESSAGE TO BE REPORTED.
```

```
2060 007306          PRINTX 0EF9302          ;REPORT OVERFLOW CONDITION.
      007306 012746 004501
      007312 012746 000001
      007316 010600
      007320 104415
      007322 062706 000004
2061 007326 004767 000002
2062 007332 000414
2063
2064 007334          50$: PRINTX 0EF9301,R5,R1,R4 ;PRINT THE MESSAGE.
      007334 010446
      007336 010146
      007340 010546
      007342 012746 004423
      007346 012746 000004
      007352 010600
      007354 104415
      007356 062706 000012
2065 007362 000207
2066 007364
      007364 004736
2067
2068 007366
      007366
      007366 104423

      JSR      PC,50$          ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
      BR       60$          ;EXIT.

      RTS      PC
      PASS

      JSR      PC,0(SP)+      ;RETURN.
                              ;RESTORE THE GPR CONTENTS.
                              ;RETURN TO PREG05 SUBRT.

      ENDMSG

L10007: TRAP      C$MSG
```

2070
2078
2079
2080
2081
2082
2083

.SBTTL GLOBAL SUBROUTINES SECTION

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

```

2085 .SBTTL GLOBAL SUBROUTINE - ALTFLD -
2086 ;* *****
2087 ;* - ALTER DEVICE REGISTER FIELDS ROUTINE -
2088 ;* THIS SUBROUTINE ALTERS THE SPECIFIED FIELD OF THE SPECIFIED DEVICE
2089 ;* REGISTER FOR THE SPECIFIED LINES. THIS ROUTINE CAN BE USED TO SET
2090 ;* OR CLEAR BITS WITHIN SELECTED FIELDS OF SELECTED REGISTERS.
2091 ;* USE EXAMPLES: SET RX.BAUD.RATE FIELDS ON LINES 3 AND 6.
2092 ;* CLEAR TX.DMA BITS ON ALL LINES.
2093 ;*
2094 ;* INPUTS: R1 - ADDRESS OF THE REGISTERS TO ALTER.
2095 ;* R2 - BIT FIELDS SET TO DESIRED STATES.
2096 ;* R3 - BIT MAP OF LINES FOR WHICH TO ALTER REGISTER.
2097 ;* R4 - MASK OF BITS TO ALTER (1 INDICATES CHANGE BIT).
2098 ;* CSRA - CONTAINS THE ADDRESS OF THE DEVICE CSR.
2099 ;* IESTAT - SAVED STATES OF THE INTERRUPT ENABLE BITS.
2100 ;*
2101 ;* OUTPUTS: DEVICE REGISTERS - SPECIFIED REGISTER FIELDS ALTERED.
2102 ;* CSR IND.ADR.REG FIELD - DESTROYED.
2103 ;*
2104 ;* CALLING SEQUENCE: JSR PC,ALTFLD
2105 ;*
2106 ;* COMMENTS: THIS ROUTINE READS THE SPECIFIED REGISTERS FOR ALL LINES
2107 ;* WITH NUMBERS LOWER THAN THE HIGHEST SPECIFIED LINE.
2108 ;* THIS ROUTINE DOES NOT READ THE CSR.
2109 ;*
2110 ;* SUBROUTINES CALLED: NONE.
2111 ;* - *****
2112
2113 007370 ALTFLD:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2114 007370 004567 174424 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2115
2116 ;*
2117 ;* SET UP TO LOOP FOR EACH LINE:
2118 ;* PREPARE THE WORD TO BE ORED INTO THE REGISTER CONTENTS.
2119 ;* SET UP THE WORD TO WRITE INTO THE IND.ADR.REG FIELD OF THE CSR.
2120 ;*
2120 007374 010400 MOV R4,R0 ;CALCULATE THE NEW CONTENTS OF THE
2121 007376 005100 COM R0 ; REGISTER FIELDS WHICH ARE TO BE
2122 007400 040002 BIC R0,R2 ; ALTERED BY THIS ROUTINE.
2123 007402 016705 172622 MOV IESTAT,R5 ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.
2124
2125 ;*
2126 ;* LOOP ONCE FOR EACH LINE, ALTERING THE SPECIFIED FIELD IN THE SPECIFIED
2127 ;* REGISTER IF THE LINE HAS BEEN SELECTED FOR ALTERING.
2128 ;* EXIT THE LOOP IF NO MORE LINES TO ALTER, OR IF WE HAVE ALTERED THE MAX
2129 ;* ALLOWABLE NUMBER OF LINES (AS SPECIFIED BY NUMLNS).
2130 ;*
2130 007406 000241 CLC ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
2131 007410 006003 ROR R3 ;GET THE LINE SELECT BIT FOR THIS LINE.
2132 007412 103006 BCC 4$ ;SKIP SETUP IF LINE IS NOT SELECTED.
2133 007414 010577 172560 MOV R5,0CSRA ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
2134 007420 011100 MOV (R1),R0 ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
2135 007422 040400 BIC R4,R0 ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
2136 007424 050200 BIS R2,R0 ;OR IN THE NEW STATES OF THE FIELDS.
2137 007426 010011 MOV R0,(R1) ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
2138 007430 005205 4$: INC R5 ;SET LINE NUMBER TO THE NEXT LINE.
2139 007432 005703 TST R3 ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
2140 007434 001365 BNE 2$ ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.

```

2141
2142 007436 60: PASS JSR ;RESTORE GPRS.
007436 004736 PC,8(SP). RETURN TO PREGOS SUBRT.
2143 007440 000207 RTS PC ;RETURN TO CALLING ROUTNE.

```
2145 .SBTTL GLOBAL SUBROUTINE
2146 ASLNTL
2147 ;* *****
2148 ;* - SETUP ASSOCIATED LINE NUMBER TABLES ROUTINE
2149 ;* THIS ROUTINE SETS UP THE TWO TABLES WHICH ARE CONTAIN INFORMATION
2150 ;* ABOUT THE TX/RX LINE WHICH IS ASSOCIATED WITH A PARTICULAR RX/TX
2151 ;* LINE. ONE TABLE IS A TABLE OF WORDS WHICH CONTAINS WORD OFFSET
2152 ;* VALUES AND THE OTHER TABLE IS A TABLE OF BYTES WHICH CONTAINS
2153 ;* LINE NUMBER VALUES.
2154 ;*
2155 ;* INPUTS: LOPBCK - STORAGE FOR THE TYPE OF LOOPBACK ON THE OUT.
2156 ;* NUMLNS - EQUATED TO THE NUMBER OF LINES ON THE OUT.
2157 ;* STGTRB - LABEL AT BASE OF STAGGERED LINE ASSOCIATION TBL.
2158 ;* TXRLNB - LABEL AT BASE OF BYTE TX/RX LINE NUMBER TABLE.
2159 ;* TXRXLB - LABEL AT BASE OF WORD TX/RX LINE NUMBER TABLE.
2160 ;* TXRXLE - LABEL AT END OF WORD TX/RX LINE NUMBER TABLE.
2161 ;*
2162 ;* OUTPUTS: TXRXL, TXRLN TABLES INITIALIZED FOR SELECTED LOOPBACK.
2163 ;*
2164 ;* CALLING SEQUENCE: JSR PC,ASLNTL
2165 ;*
2166 ;* COMMENTS:
2167 ;*
2168 ;* SUBORDINATE ROUTINES CALLED: NONE.
2169 ;* -- *****
2170 007442 ASLNTL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2171 007442 004567 174352 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2172 007446 126727 172522 000002 CMPB LOPBCK,#2 ;TEST FOR STAGGERED LOOPBACK.
2173 007454 001411 BEQ 4$ ;GO SET UP STAGGERED TABLE IF STAGGERED LPBCK.
2174 ;*
2175 ;* SET UP THE WORD TABLE FOR NON STAGGERED LOOPBACK.
2176 ;*
2177 007456 005005 CLR R5 ;CLEAR THE LINE COUNTER
2178 007460 010565 0C3710 2$: MOV R5,TXRXLB(R5) ;SET UP A WORD OF THE TABLE.
2179 007464 005205 INC R5
2180 007470 005205 INC R5 ;SET LINE COUNTER TO NEXT LINE OFFSET.
2181 007474 020527 000040 CMP R5,#2*NUMLNS ;TEST FOR ALL LINES DONE.
2182 007476 002771 BLT 2$ ;LOOP UNTIL ALL LINES DONE.
2183 000411 BR 8$ ;GO SET UP THE BYTE TABLE.
2184 ;*
2185 ;* SET UP THE WORD TABLE FOR STAGGERED LOOPBACK.
2186 ;*
2187 007500 012701 003770 4$: MOV #STGTRB,R1 ;SET UP THE SOURCE POINTER.
2188 007504 012702 003710 MOV #TXRXLB,R2 ;SET UP THE DESTINATION POINTER.
2189 007510 112122 6$: MOVB (R1)+,(R2)+ ;MOVE A BYTE INTO THE TABLE.
2190 007512 105022 CLRB (R2)+ ;CLEAR THE UPPER BYTE OF WORD TABLE ENTRY.
2191 007514 020227 003750 CMP R2,#TXRXLE ;COMPARE POINTER WITH END ADR OF TABLE.
2192 007520 002773 BLT 6$ ;LOOP IF NOT AT END YET.
2193 ;*
2194 ;* SET UP THE BYTE TABLE BASED ON THE WORD ASSOCIATION TABLE.
2195 ;*
2196 007522 012701 003710 8$: MOV #TXRXLB,R1 ;SET UP THE SOURCE POINTER.
2197 007526 012702 003750 MOV #TXRLNB,R2 ;SET UP THE DESTINATION POINTER.
2198 007532 012103 10$: MOV (R1)+,R3 ;GET THE WORD OFFSET VALUE FROM WORD TABLE.
2199 007534 006203 ASR R3 ;DIVIDE BY 2 TO GET LINE NUMBER VALUE.
2200 007536 110322 MOVB R3,(R2)+ ;LOAD THE BYTE LINE NUMBER INTO TABLE.
2200 007540 020127 003750 CMP R1,#TXRXLE ;COMPARE SOURCE POINTER WITH ADR OF TABLE END.
```

2201 007544 002772
2202
2203 007546
007546 004736
2204 007550 000207

BLT 10\$;LOOP IF NOT AT END OF TABLE YET.
60\$: PASS ;RESTORE GPRS.
RTS PC JSR PC,8(SP)+ ;RETURN TO PREG05 SUBRT.

```
2206 .SBTTL GLOBAL SUBROUTINE - CALMSL
2207 ;* *****
2208 ;* - CALIBRATE MILLI SECOND LOOP COUNT SUBROUTINE -
2209 ;* THIS SUBROUTINE CALIBRATES THE TIMING LOOP WHICH IS USED IN THE MSLOOP
2210 ;* ROUTINE. THIS SUBROUTINE CALCULATES A VALUE FOR THE MSLCNT VARIABLE
2211 ;* WHICH IS THE NUMBER OF SOFTWARE LOOPS WHICH TAKES 1 MS TO EXECUTE IN
2212 ;* THE MSLOOP ROUTINE. THIS ROUTINE CALIBRATES THE COUNT BY USING THE
2213 ;* LINE TIME CLOCK (LTC), SO IF NO LTC IS AVAILABLE THE DEFAULT VALUE FOR
2214 ;* THE DELAY COUNT MUST BE USED.
2215 ;*
2216 ;*
2217 ;* INPUTS: MSI CNT - DEFAULT 1 MS DELAY LOOP COUNT VALUE, OR
2218 ;* VALUE FROM PREVIOUS CALIBRATION.
2219 ;* MSTICK - NUMBER OF MS PER LTC CLOCK TICK.
2220 ;* TIMER1 - TIMER COUNTER CHANGED BY LTC INTERRUPT SERVICE RTN.
2221 ;* CLKHRZ - NUMBER OF LTC CLICKS PER SECOND (50 OR 60).
2222 ;*
2223 ;* OUTPUTS: CARRY - SET IF LTC IS AVAILABLE, AND NEW CALIBRATION PERFORMED.
2224 ;* MSLCNT - NEW 1 MS DELAY LOOP COUNT VALUE IF LTC AVAILABLE, OR
2225 ;* UNCHANGED IF NO LTC IS AVAILABLE.
2226 ;*
2227 ;* CALLING SEQUENCE: JSR PC,CALMSL
2228 ;*
2229 ;* COMMENTS:
2230 ;*
2231 ;* SUBORDINATE ROUTINES CALLED: UNSDIV,OOPS.
2232 ;* - *****
2233
2234 007552 CALMSL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2235 007552 004567 174242 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2236 007556 005067 000210 CLR 62$ ;CLEAR THE 2ND TIME FLAG.
2237 ;*
2238 ;* SYNCHRONIZE WITH THE LTC.
2239 ;*
2239 007562 012705 000001 2$: MOV #1,R5 ;SET OUTER LOOP COUNTER TO 1 LOOP.
2240 ;* ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE <*&
2241 ;* ;FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <*&
2242 007566 005000 CLR R0 ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
2243 007570 012767 000001 172470 MOV #1,TIMER1 ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
2244 007576 005767 172464 4$: TST TIMER1 ;CHECK FOR COUNTER HAVING GONE TO ZERO.
2245 007602 001410 BEQ 6$ ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
2246 007604 005200 INC R0 ;COUNT THIS ITERATION OF THE INNER LOOP.
2247 007606 001373 BNE 4$ ;LOOP IF COUNTER HAS NOT TURNED OVER.
2248 007610 005305 DEC R5 ;DECREMENT THE INNER LOOP COUNTER.
2249 007612 003371 BGT 4$ ;LOOP IF OUTER LOOP COUNT NOT UP.
2250 ;*
2251 ;* IF WE GOT NO LTC INTERRUPT, INDICATE THAT THERE IS NO LTC AVAILABLE.
2252 ;* LTC MUST BE FLAKEY, OR NOT REALLY AN LTC AT ALL.
2253 ;*
2254 007614 005067 172444 CLR CLKHRZ ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
2255 007620 000241 CLC ;INDICATE FAILURE FOR RETURN.
2256 007622 000461 BR 60$ ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
2257 ;*
2258 ;* WE ARE NOW SYNCHRONIZED WITH THE LTC.
2259 ;* SET UP FOR THE CALIBRATION LOOP.
2260 ;*
2261 007624 012704 002266 6$: MOV #TIMER1,R4 ;WILL TEST TIMER1 IN THE LOOP BELOW.
```

```
2262 007630 005001          CLR    R1          ;CLEAR THE OUTER LOOP COUNTER.
2263 007632 005002          CLR    R2          ;INDICATE TO CHECK ALL BITS OF TIMER1.
2264 007634 005003          CLR    R3          ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2265 007636 012714 000001    MOV    #1,(R4)    ;LOAD TIMER1 WITH COUNT OF 1.
2266
2267 007642 016705 172432    8$:    MOV    MSLCNT,R5    ;LOAD MS LOOP COUNT.
2268 007646 011400          10$:   MOV    (R4),R0    ;GET THE TIMER1 VALUE.
2269 007650 010067 000120    MOV    R0,64$    ;SAVE WORD (LIKE IN THE REAL LOOP).
2270 007654 040200          BIC    R2,R0    ;LEAVE ALL THE BITS.
2271 007656 020003          CMP    R0,R3    ;COMPARE AGAINST ZERO.
2272 007660 000261          SEC          ;SET CARRY IN CASE OF SUCCESS.
2273 007662 001406          BEQ    12$    ;EXIT LOOP IF TIMER1 HAS CLEARED.
2274 007664 005305          DEC    R5    ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2275 007666 001367          BNE    10$    ;LOOP IF MS NOT UP.
2276 007670 005301          DEC    R1    ;DECREMENT THE MS TIME COUNT.
2277 007672 001363          BNE    8$    ;KEEP LOOPING.
2278 007674 004767 000432    JSR    PC,OOPS    ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
2279
2280          ;*
2281          ; WE HAVE NOW HAVE LOOP COUNT INFORMATION FOR ONE CLOCK TICK.
2282          ; WE HAVE NEGATIVE OF NUMBER OF OUTER LOOPS IN R1, EACH IS MSLCNT INNER LOOPS.
2283          ; WE HAVE THE PORTION OF THE LAST OUTER LOOP NOT EXECUTED, IN R5.
2284          ; NOW WE CALCULATE THE TOTAL NUMBER OF INNER LOOPS EXECUTED.
2285
2285 007700 005401          12$:   NEG    R1          ;GET NUMBER OF OUTER LOOPS.
2286 007702 016702 172372    MOV    MSLCNT,R2    ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2287 007706 010203          MOV    R2,R3    ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2288 007710 160502          SUB    R5,R2    ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2289 007712 010204          MOV    R2,R4    ; AND ADD TO ACCUMULATOR LSWORD.
2290 007714 005005          CLR    R5    ;CLEAR ACCUMULATOR MSWORD.
2291 007716 005301          14$:   DEC    R1    ;CHECK R1 FOR 0 CONDITION
2292 007720 100403          BMI    16$    ;SKIP MULTIPLICATION IF ZERO
2293 007722 060304          ADD    R3,R4    ;MULTIPLY NUMBER OF INNER
2294 007724 005505          ADC    R5    ; LOOPS PER OUTER LOOP BY
2295 007726 000773          BR     14$    ;NUMBER OF OUTER LOOPS PERFORMED.
2296
2297          ;*
2298          ; DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER LTC TICK.
2299
2299 007730 016701 172342    16$:   MOV    MSTICK,R1    ;# OF MS PER LTC TICK IS DIVISOR.
2300 007734 010403          MOV    R4,R3    ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2301 007736 010502          MOV    R5,R2    ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2302 007740 004767 001200    JSR    PC,UNSDIV    ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
2303 007744 103402          BCS    18$    ;BYPASS OOPS IF WE'RE OK.
2304 007746 004767 000360    JSR    PC,OOPS    ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2305 007752 010167 172322    18$:   MOV    R1,MSLCNT    ;SET NEW VALUE FOR MS LOOP COUNT.
2306 007756 005167 000010    COM    62$    ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2307 007762 001277          BNE    2$    ;BRANCH IF ONLY ONE ITERATION DONE.
2308 007764 000261          SEC          ;SET THE SUCCESS FLAG FOR EXIT.
2309
2310 007766          60$:   PASS          ;RESTORE GPRS.
2311 007766 004736          PC,@(SP),    JSR    ;RETURN TO PREG05 SUBRT.
2312 007770 000207          RTS    PC    ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
2313
2313 007772 000000          62$:   .WORD    0    ;2ND CALIBRATION ITERATION FLAGS.
2314 007774 000000          64$:   .WORD    0    ;DUMMY WORD FOR STORAGE OF THE READ WORD.
```

```
2316 .SBTTL GLOBAL SUBROUTINE CKTRAP
2317 ;*****
2318 ;* CHECK TRAP ROUTINE -
2319 ;* THIS SUBROUTINE IS USED TO CHECK FOR A BUS TIME-OUT TRAP (004 TRAP)
2320 ;* WHICH IS CAUSED BY AN ACCESS TO A NON-EXISTENT MEMORY OR I/O LOCATION.
2321 ;* IF THE TRAP DOES NOT OCCUR, THIS ROUTINE RETURNS A SUCCESS INDICATION.
2322 ;*
2323 ;* INPUTS: R0 SOURCE ADDRESS FOR MOVE.
2324 ;* R1 - DESTINATION ADDRESS FOR MOVE.
2325 ;* (R0) - SOURCE FOR THE MOVE.
2326 ;*
2327 ;* OUTPUTS: (R1) WRITTEN TO THE CONTENTS OF (R0).
2328 ;* CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED.
2329 ;* TP4FLG - NONZERO IF TRAP OCCURRED, CLEARED OTHERWISE.
2330 ;*
2331 ;* CALLING SEQUENCE: JSR PC,CKTRAP
2332 ;*
2333 ;* COMMENTS: IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS WHICH
2334 ;* IS LABELED ADRPTR WILL BE THE TRAP PC ADDRESS ON THE STACK.
2335 ;*
2336 ;* SUBORDINATE ROUTINES CALLED: NONE.
2337 ;*****
2338
2339 007776 CKTRAP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2340 007776 004567 174016 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2341 010002 005067 172244 CLR TP4FLG ;CLEAR THE 004 TRAP FLAGS.
2342 010006 011011 MOV (R0),(R1) ;PERFORM THE MOVE IN QUESTION.
2343 010010 005767 172236 ADRPTR:: TST TP4FLG ;CHECK FOR OCCURENCE OF TRAP.
2344 010014 000261 SEC ;INDICATE SUCCESS.
2345 010016 001401 BEQ 60$ ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
2346 010020 000241 CLC ;INDICATE FAILURE.
2347 010022 004736 60$ PASS ;RESTORE GPRS.
2348 010024 000207 RTS PC JSR PC,@(SP) ;RETURN TO PREG05 SUBRT.
```

```
2349 .SBTTL GLOBAL SUBROUTINE - CLNRST -
2350 ;*****
2351 ;* - CLEAN RESET OF THE DEVICE UNDER TEST
2352 ;* THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
2353 ;* THE DUT'S SELF-TEST IS SKIPPED, AND THE FIFO IS PURGED OF ANY ERROR
2354 ;* CODES, ETC.
2355 ;* IF THE RESET DOES NOT SUCCESSFULLY COMPLETE, THEN THE CARRY BIT IS
2356 ;* PASSED BACK TO THE CALLING ROUTINE (CLEAR).
2357 ;*
2358 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR
2359 ;* TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
2360 ;* ERRNBR - ERROR NUMBER FOR POSSIBLE ERROR REPORT.
2361 ;* ERTBL - ERTYP,ERNBR,AND ERMMSG SET UP CORRECTLY.
2362 ;*
2363 ;* OUTPUTS: THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
2364 ;* CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
2365 ;* ERRLK - VALUE MAY BE DESTROYED.
2366 ;* IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
2367 ;* TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
2368 ;*
2369 ;* CALLING SEQUENCE: JSR PC,CLNRST
2370 ;*
2371 ;* COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS ERRNBR.
2372 ;* THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
2373 ;*
2374 ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET,PUFIFO,RESETT.
2375 ;*****
2376 CLNRST:: SAVE
2377 010026 004567 173766 JSR R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
2378 ;* ;CALL REGISTER SAVE SUBRT.
2379 ;*
2380 ;* RESET THE DUT.
2381 ;* THIS ROUTINE REPORTS ERRORS WITH NUMBERS FROM ERRNBR THRU ERRNBR+2.
2382 010032 004767 000604 JSR PC,RESETT ;RESET THE DUT TO A KNOWN STATE.
2383 010036 103002 BCC 60$ ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
2384 ;*
2385 ;* PURGE THE FIFO OF ERROR CODES, SAVE ANY BMP CODES FOUND.
2386 ;*
2387 010040 004767 000514 JSR PC,PUFIFO ;PURGE THE FIFO.
2388 ;*
2389 010044 60$:
2390 010044 PASS ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
2391 010044 004736 JSR PC,@(SP)+ ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
2392 010046 000207 RTS PC ;RETURN TO PREG05 SUBRT.
;CARRY BIT:IF CLEAR, THEN ABORT THE TEST.
```

```
2394 .SBTTL GLOBAL SUBROUTINE CMPMST -
2395 ;* *****
2396 ;* - COMPARE MODEM STATUS ROUTINE
2397 ;* THIS ROUTINE IS USED TO COMPARE THE PRESENT MODEM STATUS AGAINST THE
2398 ;* MODEM STATUS WHICH IS STORED IN THE MODEM STATUS STORAGE TABLE. IT
2399 ;* IGNORES THE STATES OF THE SPECIFIED SIGNALS ON A SPECIFIED LINE.
2400 ;*
2401 ;* INPUTS:      R1  LINE NUMBER OF SPECIFIED LINE.
2402 ;*              R2  - BIT MAP OF BITS TO IGNORE ON SPECIFIED LINE.
2403 ;*              CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
2404 ;*              NUMLNS EQUATED TO THE NUMBER OF LINES ON THE DUT.
2405 ;*              FSLSA - CONTAINS THE ADDRESS OF THE DUT STAT REGISTER.
2406 ;*              STSTB - LABEL AT BASE OF STAT STORAGE TABLE.
2407 ;*              TXRLNB - LABEL AT BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
2408 ;*
2409 ;* OUTPUTS:      CARRY - SUCCESS FLAG (SET IF NO DISCREPANCIES WERE FOUND).
2410 ;*
2411 ;* CALLING SEQUENCE:  JSR      PC,CMPMST
2412 ;*
2413 ;* COMMENTS:
2414 ;*
2415 ;* SUBORDINATE ROUTINES CALLED: NONE.
2416 ;* - - *****
2417
2418 010050 CMPMST:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2419 010050 004567 173744 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2420 010054 005003 ;CLEAR THE LINE COUNTER.
2421 010056 012704 002610 MOV #STSTB,R4 ;SET UP STAT STORAGE POINTER TO BASE OF TABLE.
2422 010062 010377 172112 2$: MOV R3,#CSRA ;SET UP THE CSR IND.ADR.REG FIELD.
2423 010072 011405 MOV #FSLSA,R0 ;GET THE CONTENTS OF THIS LINE'S STAT REGISTER.
2424 010074 040005 (R4),R5 ;GET THE PREVIOUS CONTENTS FROM STORAGE.
2425 010076 042400 BIC R0,R5
2426 010100 050005 BIC (R4),R0
2427 010102 012700 043777 BIS R0,R5 ;XOR PRESENT AND STORED STAT VALUES.
2428 010106 120301 MOV #43777,R0 ;PREPARE TO MASK OUT UNUSED BITS.
2429 010110 001001 CMPB R3,R1 ;TEST FOR THIS BEING SPECIFIED LINE.
2430 010112 050200 BNE 10$ ;DON'T MASK OUT SPECIFIED BITS IF IT IS NOT.
2431 010114 040005 BIS R2,R0 ;MASK OUT SPECIFIED BITS.
2432 010116 001006 10$: BIC R0,R5 ;GET BIT MAP OF UNDESIRED CHANGES.
2433 010120 005203 BNE 50$ ;EXIT WITH FAILURE IF CHANGES OCCURRED.
2434 010122 020327 000020 INC R3 ;SELECT NEXT LINE.
2435 010126 002755 CMP R3,#NUMLNS ;ALL LINES DONE?
2436 010130 000261 BLT 2$ ;LOOP IF NOT ALL LINES DONE.
2437 010132 000401 SEC ;INDICATE SUCCESS.
2438 BR 60$ ;EXIT THIS ROUTINE WITH SUCCESS.
2439 010134 000241 50$: CLC ;INDICATE FAILURE.
2440
2441 010136 60$: PASS ;RESTORE GPRS.
2442 010140 004736 000207 RTS PC JSR PC,#(SP) ;RETURN TO PREG05 SUBRT.
; CARRY - SUCCESS FLAG (SET IF SUCCESS).
```

```
2444 .SBTTL GLOBAL SUBROUTINE - DELAY
2445 ;*****
2446 ;* - DELAY SUBROUTINE
2447 ;* THIS SUBROUTINE IS USED TO DELAY A VARIABLE NUMBER OF MILLI SECONDS.
2448 ;*
2449 ;* INPUTS: R4 - CONTAINS THE NUMBER OF MS TO DELAY.
2450 ;* MSLCNT.
2451 ;*
2452 ;* OUTPUTS: NONE.
2453 ;*
2454 ;* CALLING SEQUENCE: JSR PC,DELAY
2455 ;*
2456 ;* COMMENTS: IF NO HARDWARE CLOCK INTERRUPTS ARE OCCURING, CONTROL CS WILL
2457 ;* NOT BE HONORED FOR THE DURATION OF THE DELAY.
2458 ;*
2459 ;* SUBORDINATE ROUTINES CALLED: NONE.
2460 ;*****
2461
2462 010142 DELAY:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
010142 004567 173652 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2463 010146 010401 MOV R4,R1 ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
2464 010150 012702 177777 MOV #1,R2 ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
2465 010154 005003 CLR R3 ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
2466 010156 012704 010200 MOV #62$,R4 ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
2467 010162 004767 000130 JSR PC,MSLOOP ;DELAY THE REQUESTED # OF MS.
2468 010166 103002 BCC 60$ ;EXIT ROUTINE IF WE TIMED-OUT.]
2469 010170 004767 000136 JSR PC,OOPS ;IF NO TIME OUT, BAD PROGRAM OR HOST MACHINE.
2470 010174 004736 60$: PASS ;RESTORE GPRS.
010174 000207 JSR PC,(SP) ;RETURN TO PREG05 SUBRT.
2471 010176 000207 RTS PC
2472
2473 010200 177777 62$: .WORD -1 ;DUMMY, NON-ZERO WORD.
```

```
2475 .SBTTL GLOBAL SUBROUTINE - MSLGET
2476 ;*****
2477 ;* MILLI SECONDS LOOP WHICH RETURNS READ WORD AND REMAINING TIME
2478 ;* THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
2479 ;* TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME OUT PERIOD. THE
2480 ;* CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
2481 ;* DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
2482 ;* THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
2483 ;* ROUTINE AND THEN ONCE EACH MILLI-SECOND THERE AFTER.
2484 ;* UPON RETURN, THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION
2485 ;* IS RETURNED BY THIS SUBROUTINE.
2486 ;*
2487 ;* INPUTS: R1 - TIME OUT VALUE IN MILLI SECONDS (UP TO 64K MS).
2488 ;* R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
2489 ;* R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
2490 ;* R4 - ADDRESS OF THE WORD TO TEST.
2491 ;* MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
2492 ;*
2493 ;* OUTPUTS: R0 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
2494 ;* R1 - REMAINING NUMBER OF MS IN TIME-OUT TIME.
2495 ;* CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME OUT).
2496 ;*
2497 ;* CALLING SEQUENCE: JSR PC,MSLGET
2498 ;*
2499 ;* COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
2500 ;* CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
2501 ;* ON THE SYSTEM.
2502 ;* THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
2503 ;* DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
2504 ;* LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
2505 ;* IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
2506 ;* THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
2507 ;* IF THE CONDITION IS MET, FAILURE OTHERWISE.
2508 ;*
2509 ;*
2510 ;* SUBORDINATE ROUTINES CALLED: NONE.
2511 ;*****
2512
2513 010202 MSLGET:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2514 010202 004567 173612 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2515 ;*
2516 ;* SET UP MASK FOR REMOVING UNUSED BITS IN THE TEST WORD, AND CLEAR UNUSED
2517 ;* BITS IN THE DESIRED STATE WORD TO ALLOW DIRECT COMPARISON.
2518 010206 005102 COM R2 ;GET MASK OF UNUSED BITS.
2519 010210 040203 BIC R2,R3 ;MASK OUT UNUSED BITS IN DESIRED STATE WORD.
2520 ;*
2521 ;* HANDLE THE TEST AND EXIT IF WE HAVE A 0 TIME-OUT VALUE.
2522 ;*
2523 010212 005701 TST R1 ;TEST THE TIME-OUT VALUE FOR ZERO.
2524 010214 001011 BNE 2$ ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
2525 010216 011400 MOV (R4),R0 ;GET THE WORD TO TEST BEFORE EXITING.
2526 010220 010067 000070 MOV R0,62$ ;SAVE VALUE SO WE CAN RETURN IT.
2527 010224 040200 BIC R2,R0 ;MASK OUT UNTESTED BITS OF WORD.
2528 010226 020003 CMP R0,R3 ;COMPARE AGAINST DESIRED STATE WORD.
2529 010230 000261 SEC ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
2530 010232 001420 BEQ 6$ ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
```

```
2531 010234 000241          CLC          ;INDICATE FAILURE (TIME OUT).
2532 010236 000416          BR          6$ ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
2533                          ;*
2534                          ; NON-ZERO TIME OUT VALUE. LOOP, WAITING FOR CONDITION OR TIME-OUT.
2535                          ;-
2536 010240 016705 172034    2$: MOV      MSLCNT,R5      ;LOAD MS LOOP COUNT.
2537 010244 011400          4$: MOV      (R4),R0        ;GET THE WORD TO TEST.
2538 010246 010067 000042    MOV      R0,62$          ;SAVE WORD IN CASE THIS IS THE LAST.
2539 010252 040200          BIC      R2,R0             ;MASK OUT UNTESTED BITS OF WORD.
2540 010254 020003          CMP      R0,R3             ;COMPARE AGAINST DESIRED STATE WORD.
2541 010256 000261          SEC                      ;SET CARRY IN CASE OF SUCCESS.
2542 010260 001405          BEQ      6$                ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
2543 010262 005305          DEC      R5                ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2544 010264 001367          BNE      4$                ;LOOP IF MS NOT UP.
2545 010266 005301          DEC      R1                ;DECREMENT THE MS TIME COUNT.
2546 010270 001363          BNE      2$                ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
2547 010272 000241          CLC                      ;CLEAR CARRY, WE TIMED-OUT.
2548                          ;*
2549                          ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME OUT VALUE).
2550                          ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD. EXIT ROUTINE.
2551                          ;-
2552 010274 016700 000014    6$: MOV      62$,R0        ;PASS OUT THE LAST READ WORD.
2553 010300          60$: PASS      R0,R1              ;RESTORE GPRS, EXCEPT THE FOLLOWING:
                MOV      R0,R0SLOT(SP)                ;PUT R0 IN STACK SLOT.
                MOV      R1,R1SLOT(SP)                ;PUT R1 IN STACK SLOT.
                JSR      PC,8(SP)+                    ;RETURN TO PREG05 SUBRT.
2554                          ;R0 - LAST READ WORD CHECKED FOR CONDITION.
2555                          ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
2556 010312 000207          ;CARRY SET IF SUCCESS, CLEAR IF TIME-OUT.
2557                          ;*
2558                          ; LOCAL STORAGE.
2559                          ;-
2560 010314 000000          62$: .WORD 0              ;STORAGE FOR THE LAST READ WORD.
                RTS      PC
```

```
2562 .SBTTL GLOBAL SUBROUTINE MSLOOP
2563 :*****
2564 :* - TEST LOOP SUBROUTINE -
2565 :* THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
2566 :* TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD. THE
2567 :* CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
2568 :* DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
2569 :* THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
2570 :* ROUTINE AND THEN ONCE EACH MILLI-SECOND THEREAFTER.
2571 :*
2572 :* INPUTS: R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
2573 :* R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
2574 :* R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
2575 :* R4 - ADDRESS OF THE WORD TO TEST.
2576 :* MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
2577 :*
2578 :* OUTPUTS: CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME OUT).
2579 :*
2580 :* CALLING SEQUENCE: JSR PC,MSLOOP
2581 :*
2582 :* COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
2583 :* CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
2584 :* ON THE SYSTEM.
2585 :* THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
2586 :* DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
2587 :* LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
2588 :* IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
2589 :* THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
2590 :* IF THE CONDITION IS MET, FAILURE OTHERWISE.
2591 :*
2592 :* SUBORDINATE ROUTINES CALLED: MSLGET.
2593 :*****
2594
2595 010316 MSLOOP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2596 010316 004567 173476 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2597
2598 :*
2599 :* CALLING THE MSLGET ROUTINE FROM THE MSLOOP ROUTINE ISOLATES THE CALLER OF
2600 :* MSLOOP FROM THE RETURNED TEST WORD AND REMAINING TIME-OUT VALUES.
2601 :*
2602 :* JSR PC,MSLGET ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.
2603 010322 004767 177654
2604 010326 004736 604: PASS ;RESTORE GPRS.
2604 010330 000207 JSR PC,B(CSP). ;RETURN TO PREG05 SUBRT.
;CARRY SET IF SUCCESS. CLEAR IF TIME OUT.
```

2606 .SBTTL GLOBAL SUBROUTINE OOPS
2607 ;* *****
2608 ;* - PROGRAM ABORT SUBROUTINE -
2609 ;* THIS SUBROUTINE IS USED TO ABORT THE PROGRAM WHEN A FATAL ERROR IS
2610 ;* DETECTED IN THE PROGRAM OR THE HOST SYSTEM HARDWARE. AN ERROR MESSAGE
2611 ;* IS PRINTED GIVING SOME INFORMATION ABOUT THE NATURE OF THE ABORT.
2612 ;*
2613 ;* INPUTS: R1 - ERROR CODE GIVING REASON FOR ABORT.
2614 ;*
2615 ;* OUTPUTS: AN ERROR MESSAGE IS PRINTED.
2616 ;* A LIST OF RETURN PC VALUES FOR ALL SUBROUTINE CALLS IS PRINTED.
2617 ;*
2618 ;* CALLING SEQUENCE: JSR PC.OOPS
2619 ;*
2620 ;* COMMENTS:
2621 ;*
2622 ;* SUBORDINATE ROUTINES CALLED: NONE.
2623 ;* - *****
2624
2625 010332 OOPS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
010332 004567 173462 JSR R5,PREGOS ;CALL REGISTER SAVE SUBRT.
2626 ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
2627 ERRSF 101,EM0101
010336 104454
010340 000145 TRAP C:ERSF
010342 010376 .WORD 101
010344 000000 .WORD EM0101
2628 ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL C."
2629 010346 PRINTF @EM0102 .WORD 0
010346 012746 010462 MOV @EM0102, (SP)
010352 012746 000001 MOV @1, (SP)
010356 010600 MOV SP,R0
010360 104417 TRAP C:PNTF
010362 062706 000004 ADD @4,SP
2630 010366 2\$: BREAK ;LOOK FOR OPERATOR CONTROL-C INPUT.
010366 104422 TRAP C:BRK
2631 010370 BR 2\$;INFINITE LOOP.
2632 010372 60\$: PASS ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
010372 004736 JSR PC,@(SP); RETURN TO PREGOS SUBRT.
2633 010374 000207 RTS PC ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.
2634
2635 010376 110 117 123 EM0101:: .ASCII /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED.
010401 124 040 103
010404 117 115 120
010407 125 124 105
010412 122 040 110
010415 101 122 104
010420 127 101 122
010423 105 040 117
010426 122 040 123
010431 117 106 124
010434 127 101 122
010437 105 040 102
010442 125 107 040
010445 105 116 103
010450 117 125 116
010453 124 105 122

	010456	105	104	056	
	010461	000			
2636	010462	045	116	045	EM0102:: .ASCIZ /MAPROGRAM HUNG, WAITING FOR A CONTROL C. <*****NEN/
	010465	101	120	122	
	010470	117	107	122	
	010473	101	115	040	
	010476	110	125	116	
	010501	107	054	040	
	010504	127	101	111	
	010507	124	111	116	
	010512	107	040	106	
	010515	117	122	040	
	010520	101	040	103	
	010523	117	116	124	
	010526	122	117	114	
	010531	055	103	056	
	010534	040	074	052	
	010537	052	052	052	
	010542	052	052	052	
	010545	052	052	052	
	010550	052	052	052	
	010553	045	116	045	
2637	010556	116	000		

.EVEN

```
2639 .SBTTL GLOBAL SUBROUTINE PUFIFO
2640 ;*****
2641 ;* - PURGE THE FIFO
2642 ;* THIS ROUTINE TRIES TO REMOVE ALL THE CHARACTERS FROM THE FIFO.
2643 ;* ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE BMP CODE QUEUE.
2644 ;*
2645 ;* INPUTS: RBUFA- CONTAINS THE ADDRESS OF THE RECEIVER.
2646 ;*
2647 ;*
2648 ;* OUTPUTS: CARRY BIT - INDICATES THE STATE OF THE FIFO, SET: = PURGED.
2649 ;* BMPCQ - THE CONTENTS OF THE BMP CODE QUEUE MAY BE UPDATED.
2650 ;*
2651 ;* CALLING SEQUENCE: JSR PC,PUFIFO
2652 ;*
2653 ;* COMMENTS:
2654 ;*
2655 ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
2656 ;*****
2657
2658 010560 PUFIFO::SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
010560 004567 173234 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2659 010564 012701 001000 MOV #512,,R1 ;SET MAXIMUM TRY COUNT OF 512.
2660 010570 016704 171406 MOV RBUFA,R4 ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
2661
2662 010574 011402 2$: MOV (R4),R2 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
2663 010576 100016 BPL 6$ ;EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.
2664
2665 ;*
2666 ; CHECK IF THE READ CHARACTER IS ACTUALLY A BMP CODE.
2667 ; IF IT IS, THEN SAVE IT ON THE BMP CODE QUEUE TO BE REPORTED LATER.
2668 010600 012700 070000 ;*
2669 010604 040200 MOV #70000,R0 ;GENERATE A BIT MAP OF CHAR ERROR BITS
2670 010606 001006 BIC R2,R0 ; WHICH ARE NOT SET FOR CHAR.
2671 BNE 4$ ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
2672
2673 ;*
2674 010610 012700 000301 ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
2675 010614 040200 ;*
2676 010616 001002 MOV #301,R0 ; CHECK IF BMP.
2677 010620 004767 000130 BIC R2,R0 ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
2678 BNE 4$ ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
2679 010624 005301 JSR PC,SAVBMP ;SAVE BMP CODE ON THE QUEUE.
2680 010626 001362 4$: DEC R1 ;DECREMENT THE TRY COUNT.
2681 010630 000241 BNE 2$ ;LOOP TO TRY AGAIN.
2682 010632 000401 CLC ;CLEAR CARRY, TO INDICATE FIFO NOT PURGED.
2683 010634 000261 BR 60$ ;EXIT WITH CARRY CLEAR.
2684 6$: SEC ;SET CARRY, TO INDICATE FIFO PURGED.
2685 010636 004736 60$: PASS
010636 JSR ;RESTORE GPRS,
2686 PC,@(SP), ;RETURN TO PREG05 SUBRT.
2687 010640 000207 RTS PC ;CARRY BIT, SET INDICATES FIFO PURGED.
```

```
2689 .SBTTL GLOBAL SUBROUTINE RESETT
2690 ;*****
2691 ;* - RESET DEVICE UNDER TEST *****
2692 ;* THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
2693 ;* IF RESET DOES NOT SUCCESSFULLY COMPLETE, IE. TIME-OUT OCCURS, THEN
2694 ;* AN ABORT TEST ERROR MESSAGE IS REPORTED.
2695 ;*
2696 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR
2697 ;* TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
2698 ;* ERRTBL - ERRTP,ERNBR,AND ERRMSG SET UP CORRECTLY.
2699 ;*
2700 ;* OUTPUTS: THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
2701 ;* CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
2702 ;* ERRBLK - VALUE MAY BE DESTROYED.
2703 ;* IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
2704 ;* TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
2705 ;*
2706 ;* CALLING SEQUENCE: JSR PC,RESETT
2707 ;*
2708 ;* COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERNBR
2709 ;* THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERNBR.
2710 ;*
2711 ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET.
2712 ;*****
2713
2714 010642 RESETT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2715 010642 004567 173152 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2716 010646 012702 000040 MOV #BIT05,R2 ;SET BIT MASK OF MASTER RESET BIT.
2717 ;*
2718 ;* TEST THE STATE OF THE MASTER RESET BIT IN THE CSR.
2719 ;* IF MR IS SET THEN WAIT FOR SELF-TEST TO COMPLETE.
2720 ;* IF TIME-OUT OCCURS, REPORT THE ERROR AND PASS-OUT ABORT TEST INDICATOR.
2721 ;*
2721 010652 016704 171322 MOV CSRA,R4 ;GET THE ADDRESS OF THE DUT'S CSR.
2722 010656 030214 BIT R2,(R4) ;CHECK STATE OF MASTER RESET BIT.
2723 010660 001406 BEQ 2$ ;DON'T DELAY IF MR IS ALREADY CLEAR.
2724 010662 005003 CLR R3 ;SET UP DESIRED STATE OF MASTER RESET BIT.
2725 010664 012701 011610 MOV #5000.,R1 ;PASS TIME-OUT VALUE OF 5 SECONDS.
2726 010670 004767 177306 JSR PC,MSLGET ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
2727 010674 103012 BCC 4$ ;GO REPORT ERROR IF TIMEOUT OCCURRED.
2728
2729 ;*
2730 ;* SET MASTER RESET BIT IN CSR. CLEAR TX AND RX ENABLE BITS, ETC.
2731 ;* SKIP THE SELFTEST.
2732 ;* TIME-OUT OF 5 SECS, JUST IN CASE THE SELF-TEST EXECUTES.
2733 ;*
2734 010676 010277 171276 2$: MOV R2,@CSRA ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
2735 010702 004767 000160 JSR PC,SKPSTS ;TRY TO SKIP THE SELFTEST.
2736 ;*
2737 ;* SET SELF-TEST TIME-OUT OF 5 SECONDS, AND WAIT FOR M.R TO CLEAR.
2738 ;* IF TIME-OUT OCCURS, THEN REPORT THE FATAL ERROR AND PASS-OUT THE ABORT
2739 ;* TEST INDICATOR.
2740 ;*
2741 010706 005003 CLR R3 ;SET UP DESIRED STATE OF MASTER RESET BIT.
2742 010710 012701 011610 MOV #5000.,R1 ;PASS TIME-OUT VALUE OF 5 SECONDS.
2743 010714 004767 177262 JSR PC,MSLGET ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
2744 010720 103410 BCS 6$ ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
```

```
2745
2746      ;
2747      ; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET, TEST ABORTED".
2748      ; INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
2749      ;
2750      ;
2751      ;
2752      ;
2753      ;
2754      ;
2755      ;
2756      ;
2757      ;
2758      ;
2759      ;
2760      ;
2761      ;
2762      ;
2763      ;
2764      ;
2765      ;
2766      ;
```

4\$: MOV #EM1601,R1 ;PASS ERROR MESSAGE TO REPORT.
MOV #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET 'O CLEAR'
; "TEST ABORTED"
ERROR ; >>>> ERROR <<<<
CLC ;INDICATE TEST IS TO BE ABORTED. TRAP C\$ERKOR
BR 60\$;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.

6\$: CLR IESTAT ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
SEC ;INDICATE SUCCESS, CONTINUE TEST.

60\$: PASS ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
JSR PC,@(SP), ;RETURN TO PREG05 SUBRT.
;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
RTS PC

```
2768 .SBTTL GLOBAL SUBROUTINE - SAVBMP -
2769 ;* *****
2770 ;* SAVE BMP CODES ROUTINE -
2771 ;* THIS ROUTINE SAVES THE PARAMETER PASSED IN, ONTO THE BMP CODE QUEUE
2772 ;* TOGETHER WITH THE NUMBER OF THE CURRENTLY EXECUTING TEST.
2773 ;*
2774 ;* INPUTS: R2 - CONTAINS THE BMP CODE THAT IS TO BE PLACED ON THE QUEUE.
2775 ;* BMPCQP - CONTAINS ADDRESS OF NEXT LOCATION IN THE BMP QUEUE.
2776 ;* BMPCQB - LABEL AT BASE OF THE BMP CODE QUEUE.
2777 ;* BMPCQE - LABEL OF NEXT LOCATION AFTER THE END OF THE BMP QUEUE.
2778 ;* TSTNUM - CONTAINS THE NUMBER OF THE CURRENT TEST.
2779 ;*
2780 ;* OUTPUTS: BMPCQP - INCREMENTED BY 4.
2781 ;* THE CONTENTS OF THE BMP CODE QUEUE ARE UPDATED.
2782 ;*
2783 ;* CALLING SEQUENCE: JSR PC,SAVBMP
2784 ;*
2785 ;* COMMENTS: IF THE OVERFLOW OCCURS THEN THE LAST LOCATION WILL BE
2786 ;* OVERWRITTEN BY ANY SUBSEQUENT ATTEMPTS TO UPDATE THE QUEUE.
2787 ;*
2788 ;* SUBORDINATE ROUTINES CALLED: NONE.
2789 ;*
2790 ;* *****
2791 010754 SAVBMP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2792 010754 004567 173040 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2793 010760 016704 171364 MOV BMPCQP,R4 ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
2794 010764 116724 171236 MOVB TSTNUM,(R4)+ ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
2795 010770 005204 INC R4 ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
2796 010772 042702 177400 BIC #177400,R2 ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
2797 010776 010224 MOV R2,(R4)+ ;SAVE THE BMP CODE ON THE QUEUE.
2798 011000 020427 002552 CMP R4,BMPCQE ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
2799 011004 103402 BLO 2$ ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
2800 011006 162704 000004 SUB #4,R4 ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
2801 011012 010467 171332 2$: MOV R4,BMPCQP ;SAVE THE POINTER.
2802 011016 011016 004736 60$: PASS ;RESTORE GPRS.
2803 011020 000207 RTS PC JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
```

```

2805      .SBTTL GLOBAL SUBROUTINE                      - SAVMST -
2806      ;** *****
2807      ;*
2808      ;*          - SAVE MODEM STATUS ROUTINE
2809      ;*          THIS ROUTINE SAVES THE PRESENT CONTENTS OF THE DUT STAT REGISTERS IN
2810      ;*          THE STAT STORAGE TABLE.
2811      ;*
2812      ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
2813      ;*              IESTAT - STATE OF THE DUT CSR INTERRUPT ENABLE BITS.
2814      ;*              NUMLNS - EQUATED TO THE NUMBER OF LINES ON THE DUT.
2815      ;*              FSLSA - CONTAINS THE ADDRESS OF THE DUT STAT REGISTER.
2816      ;*              STSTB - LABEL AT BASE OF THE STAT STORAGE TABLE.
2817      ;*
2818      ;* OUTPUTS:     STST TABLE - OVERWRITTEN WITH PRESENT STAT CONTENTS.
2819      ;*              CSR IND.ADR.REG FIELD - DESTROYED.
2820      ;*
2821      ;* CALLING SEQUENCE:  JSR      PC,SAVMST
2822      ;*
2823      ;* COMMENTS:      IF THE CONTENTS OF IESTAT CHANGES DURING THIS TEST THE CSR
2824      ;*                  INTERRUPT ENABLE BITS WILL NOT TRACK THE CHANGE.
2825      ;*
2826      ;* SUBORDINATE ROUTINES CALLED: NONE.
2827      ;-- *****
2828      011022 004567 172772 SAVMST:: SAVE
2829      011022 016701 171176      JSR      R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
2830      011032 012702 002610      MOV      IESTAT,R1 ;CALL REGISTER SAVE SUBRT.
2831      011036 012703 000020      MOV      STSTB,R2 ;GET IE STATES FOR UPDATING IND.ADR.REG FIELD.
2832      011042 050103      MOV      NUMLNS,R3 ;SET UP STAT STORAGE POINTER TO BASE OF TABLE.
2833      011044 010177 171130      BIS      R1,R3 ;FORM COMPLETION COMPARISON WORD.
2834      011050 017722 171132 2$: MOV      R1,@CSRA ;SET UP THE CSR IND.ADR.REG FIELD.
2835      011054 005201      MOV      @FSLSA,(R2)+ ;SAVE CONTENTS OF THIS LINE'S STAT REGISTER.
2836      011056 020103      INC      R1 ;SET LINE COUNTER TO NEXT LINE.
2837      011060 002771      CMP      R1,R3 ;CHECK FOR ALL LINES DONE.
2838      011062      BLT      2$ ;LOOP IF NOT ALL LINES DONE.
2839      011062 004736 60$: PASS      JSR      PC,@(SP)+ ;RESTORE GPRS.
2840      011064 000207      RTS      PC ;RETURN TO PREG05 SUBRT.

```

```
2842 .SBTTL GLOBAL SUBROUTINE - SKPSTS
2843 ;* *****
2844 ;* - SKIP SELFTEST ROUTINE
2845 ;* THIS SUBROUTINE IS USED TO SKIP THE SELFTEST AFTER A DUT RESET HAS BEEN
2846 ;* INITIATED. IT MUST BE ENTERED IMMEDIATELY AFTER SETTING THE DUT MASTER
2847 ;* RESET ROUTINE OR AFTER THE EXECUTION OF A BUS RESET (BECAUSE OF TIMING
2848 ;* CONSIDERATIONS).
2849 ;*
2850 ;* INPUTS: CSRA - CONTAINS ADDRESS OF THE DUT CSR.
2851 ;* TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
2852 ;*
2853 ;* OUTPUTS: SKIP SELFTEST CODES ARE WRITTEN TO THE DUT REGISTERS.
2854 ;*
2855 ;* CALLING SEQUENCE: JSR PC,SKPSTS
2856 ;*
2857 ;* COMMENTS:
2858 ;*
2859 ;* SUBORDINATE ROUTINES CALLED: DELAY.
2860 ;* - *****
2861
2862 011066 SKPSTS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2863 011066 004567 172726 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2864 011072 012704 000012 MOV #10.,R4 ;PASS DELAY VALUE OF 10 MILLI SECONDS.
2865 011076 004767 177040 JSR PC,DELAY ;DELAY FOR 10 MILLI-SECONDS.
2866 ;*
2867 ;* WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
2868 ;*
2869 ;* MOV #NUMLNS!BIT05,R1 ;FORM IND.ADR.REG FIELD (PLUS M.R. BIT) WORD.
2870 ;* THE ABOVE INCLUSION OF THE M.R. BIT IS NECESSARY BECAUSE OF THE
2871 ;* LACK OF A M.R. BIT WRITE LOCK-OUT ON THE DMU-11.
2872 011106 012703 052525 MOV #52525,R3 ;INITIALISE THE SKIP SELF-TEST CODE.
2873 011112 005301 4$: DEC R1 ;SELECT THE NEXT SET OF DEVICE REGISTERS.
2874 011114 016704 171060 MOV CSRA,R4 ;GET THE ADDRESS OF THE CSR OF THE DUT.
2875 011120 010124 6$: MOV R1,(R4)+ ;SELECT A BANK OF DUT REGISTERS.
2876 011122 010324 MOV R3,(R4)+ ;WRITE THE CODE TO A DUT REGISTER.
2877 011130 103774 CMP R4,TXBFCA ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
2878 011132 032701 000017 BLO 6$ ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
2879 011136 001365 BIT #17,R1 ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
2880 BNE 4$ ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.
2881 011140 60$: PASS ;RESTORE GPRS.
2882 011140 004736 JSR PC,@(SP). ;RETURN TO PREG05 SUBRT.
2882 011142 000207 RTS PC
```

```
2884 .SBTTL GLOBAL SUBROUTINE
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907 011144
011144 004567 172650
2908
2909
2910
2911 011150 010204
2912 011152 160104
2913 011154 103403
2914 011156 012701 177777
2915 011162 000442
2916
2917
2918
2919 011164 005004
2920 011166 000241
2921 011170 006001
2922 011172 006004
2923 011174 012700 000020
2924
2925
2926
2927 011200 010246
2928 011202 010346
2929 011204 160403
2930 011206 005602
2931 011210 103402
2932 011212 160102
2933 011214 103003
2934
2935
2936
2937
2938 011216 012603
2939 011220 012602

.SBTTL GLOBAL SUBROUTINE
UNSDIV -
*****
;*
;*      UNSIGNED DIVIDE ROUTINE
;*      THIS SUBROUTINE IS USED TO DIVIDE A 32 BIT UNSIGNED DIVIDEND BY A
;*      16 BIT UNSIGNED DIVISOR GIVING A 16 BIT QUOTIENT. ALL NUMBERS ARE
;*      CONSIDERED TO BE UNSIGNED. A SUCCESS FLAG IS NOT SET ON RETURN IF
;*      THE QUOTIENT WAS TOO BIG TO BE CONTAINED IN 16 BITS.
;*
;* INPUTS:      R1 - THE DIVISOR, UNSIGNED, 16 BITS.
;*              R2 - MOST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
;*              R3 - LEAST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
;*
;* OUTPUTS:     R1  QUOTIENT, UNSIGNED, 16 BITS (177777 IF OVERFLOW).
;*              CARRY SUCCESS FLAG, SET IF COMPLETE QUOTIENT FITS IN 16 BITS.
;*
;* CALLING SEQUENCE:  JSR      PC,UNSDIV
;*
;* COMMENTS:      IF THE DIVISOR IS 0 THE QUOTIENT IS RETURNED AS ALL ONES
;*                  (177777) AND THE CARRY IS CLEAR REGARDLESS OF THE DIVIDEND.
;*
;* SUBORDINATE ROUTINES CALLED: NONE.
;*
;* *****
UNSDIV:: SAVE
;SAVE CONTENTS OF GPRS R0 THRU R5.
JSR      R5,PREG05 ;CALL REGISTER SAVE SUBRT.
;*
;* CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
;*
;*      MOV      R2,R4 ;GET MSW OF DIVIDEND FOR SUBTRACT.
;*      SUB      R1,R4 ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
;*      BCS      2$ ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
;*      MOV      0 1,R1 ;SET QUOTIENT TO ALL ONES (177777).
;*      BR       60$ ;EXIT WITH CARRY CLEAR.
;*
;* SET UP COUNTERS AND VARIOUS WORKING GPRS.
;*
2$:      CLR      R4 ;CLEAR THE LSW OF THE DIVISOR.
;*      CLC ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
;*      ROR      R1 ;DIVISOR BY
;*      ROR      R4 ;2(UNSIGNED)
;*      MOV      0 16.,R0 ;SET UP INITIAL SHIFT COUNT TO 16.
;*
;* THE SUBTRACT AND SHIFT LOOP.
;*
4$:      MOV      R2,-(SP) ;SAVE MSWORD OF DIVIDEND.
;*      MOV      R3,(SP) ;SAVE LSWORD OF DIVIDEND.
;*      SUB      R4,R3 ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
;*      SBC      R2 ;MSWORD DIVIDEND - BORROW
;*      BCS      6$ ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
;*      SUB      R1,R2 ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
;*      BCC      8$ ;IF NO BORROW, IT WENT, CARRY IS CLEAR.
;*
;* IT DIDN'T GO, SO WE SHIFT A 1 INTO THE QUOTIENT (COMPLEMENTED LATER).
;* CARRY IS SET.
;*
6$:      MOV      (SP)+,R3 ;RESTORE LSWORD OF DIVIDEND.
;*      MOV      (SP)+,R2 ;RESTORE MSWORD OF DIVIDEND.
```

```
2940 011222 000401      BR      10$      ;GOTO SHIFT 1 INTO THE QUOTIENT.
2941      ;*
2942      ; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
2943      ; COMPLEMENTED LATER).  CARRY IS CLEAR.
2944      ;
2945 011224 012626      8$:  MOV      (SP)+,(SP)+      ;POP THE SAVED DIVIDEND OFF OF THE STACK.
2946      ;*
2947      ; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.
2948      ;-
2949 011226 006105      10$:  ROL      R5      ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
2950 011230 000241      CLC      ;DIVIDE THE
2951 011232 006001      ROR      R1      ; DEVISOR BY
2952 011234 006004      ROR      R4      ; 2 (UNSIGNED).
2953 011236 005300      DEC      R0      ;COUNT THIS SHIFT AND SUBTRACT.
2954 011240 001357      BNE      4$      ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
2955 011242 005105      COM      R5      ;GET QUOTIENT FROM INVERTED QUOTIENT.
2956      ;*
2957      ; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.
2958      ;-
2959 011244 000241      CLC      ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
2960 011246 006103      ROL      R3      ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
2961 011250 103402      BCS      12$      ;IF CARRY FROM SHIFT, ROUND UP.
2962 011252 160403      SUB      R4,R3      ;SUBTRACT DIVISOR FROM DIVIDEND.
2963 011254 103403      BCS      14$      ;IF BORROW, DON'T ROUND UP.
2964      ;*
2965      ; ROUND UP, EXTRA SUBTRACT WENT.
2966      ;-
2967 011256 005205      12$:  INC      R5      ;INCREMENT THE QUOTIENT BY ONE.
2968 011260 001001      BNE      14$      ;IF NO OVERFLOW, WE LEAVE THE ROUND UP
2969 011262 005305      DEC      R5      ;DON'T LET ROUNDING CAUSE OVERFLOW.
2970      ;*
2971      ; ALL DONE, PASS QUOTIENT AND EXIT.
2972      ;-
2973 011264 010501      14$:  MOV      R5,R1      ;PASS QUOTIENT BACK IN R1.
2974 011266 000261      SEC      ;INDICATE NO OVERFLOW.
2975      ;
2976 011270      60$:  PASS      R1      ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
      011270 010166 000004      MOV      R1,R1SLOT(SP)      ;PUT R1 IN STACK SLOT.
      011274 004736      PC,8(SP)+      ;RETURN TO PREG05 SUBRT.
2977      ;R1 - 16 BIT, UNSIGNED QUOTIENT.
2978 011276 000207      RTS      PC      ;CARRY - SET INDICATES NO OVERFLOW (SUCCESS).
```

```
2980 .SBTTL GLOBAL SUBROUTINE WAIBIC
2981 ;** *****
2982 ;* - WAIT FOR BIT CLEAR ROUTINE
2983 ;* THIS SUBROUTINE WAITS FOR THE SPECIFIED BIT TO BECOME CLEAR. IF THE
2984 ;* SPECIFIED BIT GOES TO A CLEAR STATE WITHIN THE SPECIFIED TIME OUT
2985 ;* PERIOD A SUCCESS INDICATION IS RETURNED BY THIS ROUTINE.
2986 ;* THE LAST VALUE WHICH IS READ LOOKING FOR THE CONDITION IS RETURNED TO
2987 ;* ALLOW THE USE OF THIS ROUTINE TO LOOK FOR DESTRUCTIVE READ CONDITIONS.
2988 ;*
2989 ;* INPUTS: R1 TIME-OUT VALUE AND BIT NUMBER INDICATION:
2990 ;* BITS 15 THRU 12 NUMBER OF BIT TO TEST (RANGE 0 THRU 15).
2991 ;* BITS 11 THRU 0 TIME OUT VALUE IN MILLI-SECONDS (4095 MAX).
2992 ;* R2 - ADDRESS OF WORD CONTAINING THE BIT TO TEST.
2993 ;* MSLCNT.
2994 ;*
2995 ;* OUTPUTS: R2 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
2996 ;* CARRY - SUCCESS FLAG (CARRY SET IF BIT CLR BEFORE TIME OUT).
2997 ;*
2998 ;* CALLING SEQUENCE: MOV #130040,R1 ;PASS BIT 11 (13 OCTAL) AND
2999 ;* ; 32 (40 OCTAL) MS DELAY.
3000 ;* MOV #LABEL,R2 ;TEST BIT IN WORD AT "LABEL".
3001 ;* JSR PC,WAIBIC ;WAIT 32 MS FOR BIT 11 TO CLR.
3002 ;*
3003 ;* COMMENTS:
3004 ;*
3005 ;* SUBORDINATE ROUTINES CALLED: MSLGET.
3006 ;*
3007 ;* - - - - -
3008 011300 WAIBIC:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3009 011300 004567 172514 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3010 011304 010204 ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
3011 011310 042701 170000 MOV R2,R4
3012 011314 042702 007777 MOV R1,R2
3013 011320 000302 BIC #170000,R1 ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
3014 011322 006202 BIC #7777,R2 ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
3015 011324 006202 SWAB R2 ;PUT LINE NUMBER FIELD IN LS BYTE.
3016 011326 006202 ASR R2 ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
3017 011330 016202 002310 ASR R2 ; POSITION TO USE IT AS A WORD TABLE OFFSET
3018 011334 005003 MOV BITTBL(R2),R2 ; FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
3019 011336 004767 176640 CLR R3 ;GET BIT MAP OF LINE TO TEST FROM TABLE.
3020 JSR PC,MSLGET ;INDICATE THAT THE BIT SHOULD BE CLR.
3021 011342 010002 ;WAIT FOR THE BIT TO BE CLR WITHIN TIME-OUT.
3022 011344 010266 000006 604: MOV R0,R2 ; CARRY IS CORRECT UPON MSLGET RETURN.
3023 011350 004736 PASS R2 ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
3024 011352 000207 RTS PC ;RESTORE GPRS. EXCEPT THE FOLLOWING:
; R2,R2SLOT(SP) ;PUT R2 IN STACK SLOT.
; PC,8(SP) ;RETURN TO PREG05 SUBRT.
; R2 - LAST VALUE READ LOOKING FOR CONDITION.
; CARRY - SUCCESS FLAG (SET IF BIT FOUND CLR).
```

```
3026 .SBTTL GLOBAL SUBROUTINE
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 011354 004567 172440
3055 011354 010204
3056 011362 010102
3057 011364 042701 170000
3058 011370 042702 007777
3059 011374 000302
3060 011376 006202
3061 011400 006202
3062 011402 006202
3063 011404 016202 002310
3064 011410 010203
3065 011412 004767 176564
3066
3067 011416 010002
3068 011420
3069 011420 010266 000006
3070 011424 004736
3070 011426 000207

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

WAIBIS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
; R5,PREG05 ;CALL REGISTER SAVE SUBRT.
;SET UP THE ADDRESS PARAMETER FOR MSLGET.
MOV R2,R4
MOV R1,R2
BIC #170000,R1
BIC #7777,R2
;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
;PUT LINE NUMBER FIELD IN LSBYTE.
SWAB R2
;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
; POSITION TO USE IT AS A WORD TABLE OFFSET
ASR R2
; FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
ASR R2
MOV BITTBL(R2),R2
;GET BIT MAP OF LINE TO TEST FROM TABLE.
MOV R2,R3
;INDICATE THAT THE BIT SHOULD BE SET.
JSR PC,MSLGET
;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
; CARRY IS CORRECT UPON MSLGET RETURN.
;PASS LAST VALUE READ AS OUTPUT PARAMETER.
;RESTORE GPRS, EXCEPT THE FOLLOWING:
MOV R2,R2SLOT(SP) ;PUT R2 IN STACK SLOT.
PC,0(SP)+ ;RETURN TO PREG05 SUBRT.
; R2 - LAST VALUE READ LOOKING FOR CONDITION.
; CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).
```

```
3072 .SBTTL GLOBAL SUBROUTINE                                WTWLNC
3073 ;* .....
3074 ;* - LINE CONTROL REGISTER SETUP ROUTINE
3075 ;* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
3076 ;* CONTROL REGISTERS (LNCTRL) TO THE SPECIFIED STATE. ONLY THE LNCTRLS
3077 ;* FOR THE SPECIFIED LINES ARE ALTERED.
3078 ;*
3079 ;* INPUTS:      R0  NEW LINE PARAMETERS.
3080 ;*              R5  BIT MAP OF LINES TO BE ALTERED.
3081 ;*              CSRA - CONTAINS ADDRESS OF THE DUT CSR.
3082 ;*              IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
3083 ;*              ENABLE BITS IN THE CSR.
3084 ;*              LNCTRA - CONTAINS ADDRESS OF THE DUT LNCTRL REGISTERS.
3085 ;*
3086 ;* OUTPUTS:     LNCTRL - SPECIFIED DUT LINE CONTROL REGISTERS ARE ALTERED.
3087 ;*
3088 ;* CALLING SEQUENCE:  JSR      PC,WTWLNC
3089 ;*
3090 ;* COMMENTS:
3091 ;*
3092 ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
3093 ;* .....
3094
3095 011430 WTWLNC:: SAVE
3096 011430 004567 172364                JSR      R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
3097 ;*                                     ;CALL REGISTER SAVE SUBRT.
3098 ;* SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
3099 ;*
3099 011434 016701 170550                MOV      LNCTRA,R1 ;SET UP THE REGISTER ADDRESS PARAMETER.
3100 011440 010002                MOV      R0,R2 ;SET UP THE DESIRED REGISTER CONTENTS.
3101 011442 010503                MOV      R5,R3 ;SET UP THE BIT MAP OF LINES TO ALTER.
3102 011444 012704 177777                MOV      #1,R4 ;SELECT ALL REGISTER BITS TO BE ALTERED.
3103 ;*
3104 ;* CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
3105 ;*
3106 011450 004767 175714                JSR      PC,ALTFLD ;ALTER THE REGISTER CONTENTS.
3107 ;*
3108 011454 ;RESTORE GPRS.
3108 011454 004736                PASS
3109 011456 000207                RTS      PC JSR      PC,@(SP). ;RETURN TO PREG05 SUBRT.
```

```

3111 .SBTTL INTERRUPT SERVICE ROUTINE          CLKINT
3112 ;* *****
3113 ;* THIS ROUTINE IS EXECUTED CLKHRZ TIMES PER SECOND. IT DECREASES THE
3114 ;* TWO TIMER COUNTERS DOWN TO ZERO.
3115 ;*
3116 ;* INPUTS:      TIMER1 - TIMER COUNTER #1.
3117 ;*              TIMER2 - TIMER COUNTER #2.
3118 ;*              TIMER3 - TIMER COUNTER FOR CALL OF BREAK MACRO.
3119 ;*
3120 ;* OUTPUTS:     THE 2 TIMER COUNTERS ARE DECREMENTED IF THEY ARE NOT ZERO.
3121 ;*
3122 ;* CALLING SEQUENCE: PUT #CLKINT IN THE CLOCK INTERRUPT VECTOR SLOT.
3123 ;*                  PUT THE DESIRED TIME PERIOD (SECONDS TIMES CLKHRZ) IN
3124 ;*                  EITHER TIMER1 OR TIMER2 AND POLL THE RESPECTIVE TIMER
3125 ;*                  COUNTER TO DETECT ITS GOING TO 0 ON TIME OUT.
3126 ;*
3127 ;* COMMENTS:    THE 2 COUNTERS WILL NOT WRAPAROUND BUT WILL STOP AT 0. THIS
3128 ;*              ALLOWS THE DETECTION OF A TIME-OUT ANY TIME AFTER THE TIME OUT
3129 ;*              HAS OCCURRED UNTIL THE TIMER COUNTER IS SET TO ANOTHER VALUE.
3130 ;*
3131 ;* SUBORDINATE ROUTINES CALLED: NONE.
3132 ;* - - *****
3133
3134 011460 005767 170602 CLKINT:: TST    TIMER1      ;CHECK FOR TIMER1 AT ZERO.
3135 011464 001402      BEQ     2$      ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
3136 011466 005367 170574      DEC     TIMER1    ;DECREMENT TIME COUNT.
3137 011472 005767 170572 2$: TST    TIMER2    ;CHECK FOR TIMER2 AT ZERO.
3138 011476 001402      BEQ     4$      ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
3139 011500 005367 170564      DEC     TIMER2    ;DECREMENT TIME COUNT.
3140 011504 005367 170562 4$: DEC     TIMER3    ;DECREMENT THE BREAK COUNT.
3141 011510 001006      BNE     60$      ;EXIT IF NOT TIME TO CALL BREAK.
3142 011512 016767 170556 170552 MOV    BCOUNT,TIMER3 ;SET UP TIME TILL NEXT BREAK.
3143 011520 010046      MOV    RO,-(SP)    ;SAVE CONTENTS OF RO FROM BREAK MACRO.
3144 011522      BREAK      ;CHECK FOR OPERATOR CONTROL/C.
3145 011524 012600      TRAP    C$BRK
3146 011526 000002 60$: MOV    (SP)+,RO    ;RESTORE CONTENTS OF RO.
                       RTI

```

```
3148 .SBTTL GLOBAL TRAP SERVICE ROUTINE TP4RTN
3149 ;*****
3150 ;* BUS TIME-OUT TRAP (004 TRAP) SERVICE ROUTINE
3151 ;* THIS ROUTINE DETERMINES IF THE 004 TRAP WAS CAUSED BY
3152 ;* AN "EXPECTED" ERROR OR NOT BY EXAMINING THE RETURN PC VALUE ON THE
3153 ;* STACK. IF THE TRAP IS UNEXPECTED, THIS ROUTINE JUMPS TO THE NORMAL
3154 ;* DIAGNOSTIC SUPERVISOR 004 TRAP HANDLING ROUTINE.
3155 ;*
3156 ;*
3157 ;* INPUTS: SP - POINTS TO THE PC WHERE THE TRAP OCCURED.
3158 ;* ADRPTR - LABEL AT THE ADDRESS WHERE "EXPECTED" TRAPS OCCUR.
3159 ;* TP4FLG 004 TRAP FLAGS.
3160 ;*
3161 ;* OUTPUTS: TP4FLG - BIT 15 IS SET IF "EXPECTED" TRAP OCCURED.
3162 ;*
3163 ;* CALLING SEQUENCE: PUT ADDRESS POINTED TO BY TP4RTN IN 004 VECTOR.
3164 ;* OCCURENCE OF 004 TRAP VECTORS TO THIS ROUTINE.
3165 ;*
3166 ;* COMMENTS: ANY 004 TRAP WHICH OCCURS AT AN ADDRESS OTHER THAN THAT LABELED
3167 ;* ADRPTR WILL BE HANDLED BY THE NORMAL 004 TRAP SERVICE ROUTINE.
3168 ;*
3169 ;* SUBORDINATE ROUTINES CALLED: NONE.
3170 ;*****
3171
3172 011530 021627 010010 TP4RTN:: CMP (SP),#ADRPTR ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
3173 011534 001402 BEQ 2$ ;IF THEY MATCH, CONTINUE THIS ROUTINE.
3174 011536 000177 170506 JMP 8TP4VEC ;IF NOT, JUMP TO NORMAL 004 TRAP SERVICE RTN.
3175 011542 052767 100000 170502 2$: BIS #BIT15,TP4FLG ;SET THE 004 TRAP OCCURED FLAG.
3176 011550 000002 RTI ;ALL DONE, GO BACK TO THE TEST.
```

3185
3186
3187
3188
3189
3190
3191
3192
3193

3194 011552
011552

3195

3196 011552
011552 000167
011554 000000

3197

3198

3199

3200 011556
011556

011556 104425

.SBTTL REPORT CODING SECTION

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

BGNRPT

EXIT RPT

.EVEN

ENDRPT

L\$RPT::

.WORD J\$JMP
.WORD L10010-2-.

L10010:
TRAP C\$RPT

```
3202
3210 .SBTTL PROTECTION TABLE
3211
3212
3213 ;**
3214 ; THIS TABLE IS USED BY THE RUNTIME SERVICES
3215 ; TO PROTECT THE LOAD MEDIA.
3216 ;
3217 011560 BGNPROT
      011560
3218
3219 011560 177777 L$PROT::
3220 011562 177777      1 ;OFFSET INTO P TABLE FOR CSR ADDRESS
3221 011564 177777     -1 ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
3222      -1 ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
3223 011566
3224 ENDPROT
```

```

3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260 011566
      011566
3261
3262 011566
      011566 012700 000040
      011572 104447
3263 011574
      011574 103416
3264
3265 011576
      011576 012700 000037
      011602 104447
3266 011604
      011604 103556
3267
3268 011606
      011606 012700 000035
      011612 104447
3269 011614
      011614 103555
3270
3271 011616
      011616 012700 000036
      011622 104447
3272 011624
      011624 103161
3273 011626 000167 000522
3274 011632
3275 011632
      011632 104433
3276
3277
3278
3279 011634
      011634 012700 000114
      011640 104462
      011642 010001
3280 011644 012167 170406
3281 011650 012167 170404
3282 011654 012167 170402
3283 011660 012167 170400
3284 011664 026727 170374 000062
3285 011672 001004

```

```

3286 011674 012767 000024 170374      MOV    #20.,MSTICK      ;INDICATE 20MS PER CLOCK TICK.
3287 011702 000403                    BR      4$
3288 011704 012767 000021 170364 2$:  MOV    #17.,MSTICK      ;INDICATE 17 MS PER CLOCK TICK.
3289 011712 012746 000300 4$:  SETVEC  CLKVEC,#CLKINT,#PRI06 ;INITIALIZE CLOCK INTERRUPT VECTOR.
                                MOV    #PRI06,-(SP)
                                MOV    #CLKINT,-(SP)
                                MOV    CLKVEC,-(SP)
                                MOV    #3,-(SP)
                                TRAP    C$SVEC
                                ADD     #10,SP
3290 011740 016700 170320      MOV    CLKHRZ,R0      ;INITIALIZE THE BREAK COUNT
3291 011744 006300              PSL     R0              ; TO CAUSE A BREAK
3292 011746 010067 170322      MOV    R0,BCOUNT      ; EVERY 2 SECONDS.
3293 011752 012700 000240      SETPRI  #PRI05      ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
                                MOV     #PRI05,R0
                                TRAP    C$SPRI
3294
3295 ;*
3296 ; ENABLE THE LINE TIME CLOCK (LTC) CHECKING TO MAKE SURE THAT THE CSR
3297 ; IS ACCESSABLE.
3298 ; FIRST SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
3299 011760 016767 166020 170262      MOV    4,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
3300 011766 012767 011530 166010      MOV    #TP4RTN,4      ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
3301
3302 ;*
3303 ; ENABLE LTC CHECKING FOR 004 TRAP IN CASE CSR IS NOT THERE.
3304 011774 005067 170252      CLR     TP4FLG      ;CLEAR THE 004 TRAP FLAG.
3305 012000 012767 000100 170246      MOV    #BIT6,WORD1      ;SET UP TO SET BIT6 OF THE LTC CSR.
3306 012006 012700 002254      MOV    #WORD1,R0      ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
3307 012012 016701 170240      MOV    CLKCSR,R1      ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
3308 012016 004767 175754      JSR     PC,CKTRAP      ;MOVE AND CHECK FOR TRAP.
3309 012022 016767 170222 165754      MOV    TP4VEC,4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
3310 012030 103403              BCS     6$              ;IF NO TRAP, LTC IS THERE SO CONTINUE.
3311 012032 005067 170226      CLR     CLKHRZ      ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
3312 012036 000402              BR      8$              ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
3313
3314 ;*
3315 ; CALIBRATE THE DELAY ROUTINE MILLI-SECOND DELAY COUNT VALUE.
3316 012040 004767 175506      6$:  JSR     PC,CALMSL
3317
3318 ;*
3319 ; CHECK FOR MEMORY MANAGEMENT PRESENT ON THIS MACHINE.
3320 ; IF MEM MGT IS PRESENT, DISABLE IT.
3321 012044 016767 165734 170176      8$:  MOV    4,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
3322 012052 012767 011530 165724      MOV    #TP4RTN,4      ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
3323 012060 005067 170166      CLR     TP4FLG      ;CLEAR THE 004 TRAP FLAG.
3324 012064 005067 170164      CLR     WORD1      ;PREPARE TO CLEAR THE MEM MGT SRO REGISTER.
3325 012070 012700 002254      MOV    #WORD1,R0      ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
3326 012074 016701 170202      MOV    MMSRO,R1      ;SELECT MEM MGT SRO REGISTER AS DESTINATION.
3327 012100 005067 170200      CLR     MMPRES      ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
3328 012104 005067 170176      CLR     MMENAB      ;INDICATE MEM MGT IS NOT ENABLED.
3329 012110 004767 175662      JSR     PC,CKTRAP      ;CLEAR THE MEM MGT SRO REG AND CHECK FOR TRAP.
3330 012114 016767 170130 165662      MOV    TP4VEC,4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
3331 012122 103003              BCC     10$             ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
3332 012124 012767 000001 170152      MOV    #1,MMPRES      ;INDICATE THAT MEM MGT IS PRESENT.
3333 012132 005067 170100      10$:  CLR     PASCNT      ;CLR COUNTER USED IN REPORTING ROM VERSION #.
3334 012136 000167 000006      JMP     NEWPAS      ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.

```

```

3335
3336 012142          NEWRES: BRESET                ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
      012142 104433          ;TRAP C$RESET
3337 012144 005067 170066          CLR PASCNT        ;CLR COUNTER USED IN REPORTING ROM VERSION #.
3338 012150          NEWPAS:
3339 012150 012767 177777 170020  MOV # 1,UNITN      ;RESET LOGICAL DEVICE TO -1
3340
3341          ;+
3342          ; INCREMENT THE PASS COUNTER, CORRECT FOR ANY OVERFLOW.
3343          ; THIS COUNTER IS USED IN THE ROM VERSION TEST.
3344 012156 005267 170054          INC PASCNT        ;INCREMENT THE PASS COUNTER.
3345 012162 001002          BNE GETPRM              ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
3346 012164 005367 170046          DEC PASCNT        ;SET PASS COUNT TO 177777 OCTAL.
3347
3348          ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
3349 012170          GETPRM:
3350 012170 005267 170002          INC UNITN          ;INCREMENT LOGICAL DEVICE NUMBER
3351 012174 026767 167776 167610  CMP UNITN,L$UNIT  ;SEE IF MAXIMUM UNIT NO. EXCEEDED
3352 012202 002362          BGE NEWPAS              ;BR IF YES
3353
3354 012204          GPHARD UNITN,R1                ;GET P TABLE POINTER INTO R1
      012204 016700 167766
      012210 104442
      012212 010001
3355 012214          BCOMPLETE 30$                ;BR IF DEVICE AVAILABLE
      012214 103401
3356 012216 000764          BR GETPRM              ;SKIP THIS DEVICE
3357
3358
3359          ;***** HARDWARE PARAMETER MOVING CODE *****
3360 012220 012167 167754 30$: MOV (R1)+,CSRA      ;STORE DMU-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
3361 012224 012167 167742      MOV (R1)+,ACTLNS    ;STORE DMU-11 ACTIVE LINE BIT MAP
3362 012230 111167 167740      MOVB (R1),LOPBCK    ;STORE DMU 11 LOOPBACK MODE
3363
3364          ;+
3365          ; CALCULATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
3366          ; DEVICE REGISTER ADDRESS TABLE.
3367 012234 016701 167740          MOV CSRA,R1        ;COPY CSR ADDRESS
3368 012240 005201          INC R1                  ;INCREMENT CSR ADDRESS
3369 012242 005201          INC R1                  ; COPY BY 2.
3370 012244 012703 000007      MOV #7,R3          ;SET UP REGISTER COUNT
3371 012250 012702 002202      MOV #RBUFA,R2       ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
3372 012254 010122 12$: MOV R1,(R2)+             ;STORE REGISTER ADDRESS IN TABLE
3373 012256 005201          INC R1                  ;INCREMENT REGISTER ADDRESS
3374 012260 005201          INC R1                  ; BY 2,FOR THE NEXT DEVICE REGISTER.
3375 012262 005303          DEC R3                  ;DECREMENT REGISTER COUNT
3376 012264 001373          BNE 12$                ;LOOP IF NOT DONE
3377
3378          ;+
3379          ; INITIALISE THE BMP CODE QUEUE.
3380
3381 012266 012700 002352          MOV #BMPCQB,R0      ;GET THE START ADDRESS OF THE QUEUE.
3382 012272 012701 002552          MOV #BMPCQE,R1      ;GET THE END ADDRESS OF THE QUEUE.
3383 012276 010067 170046          MOV R0,BMPCQP      ;SET THE POINTER TO THE START OF THE QUEUE.
3384 012302 005020 14$: CLR (R0)+                 ;CLEAR OUT THE CONTENTS OF THE QUEUE.
3385 012304 020001          CMP R0,R1                ;CHECK IF END OF QUEUE HAS BEEN REACHED.
3386 012306 103775          BLO 14$                 ;LOOP IF NOT ALL DONE.

```

```

3387
3388
3389
3390
3391 012310 032767 000020 167644
3392 012316 001416
3393 012320 026727 167466 000001
3394 012326 003412
3395 012330
    012330 016746 167642
    012334 012746 004601
    012340 012746 000002
    012344 010600
    012346 104417
    012350 062706 000006
3396 012354
3397
3398 012354 005067 167642
3399
3400
3401
3402 012360
    012360 012700 000340
    012364 104441
3403 012366
    012366
    012366 104411
3404
3405 000000

;
; REPORT THE UNIT NUMBER IF THE SOFTWARE P TABLE QUESTION WAS ANSWERED YES.
; AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
;
    BIT    @BIT4,OPTION    ;CHECK IF THE QUESTION WAS ANSWERED YES.
    BEQ    16$             ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
    CMP    L$UNIT,#1       ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
    BLE    16$             ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
    PRINTF @MFUNIT,UNITN   ;REPORT UNIT NUMBER.

                                MOV    UNITN, -(SP)
                                MOV    @MFUNIT, (SP)
                                MOV    @2, (SP)
                                MOV    SP, R0
                                TRAP    C$PNTF
                                ADD     @6, SP
16$:
ENDIT: CLR    CTRLCF        ;CLR THE CTRL C TEST ABORT FLAG.
;
; SET THE PROCESSOR PRIORITY TO DISABLE ALL INTERRUPTS.
;
    SETPRI @PRI07           ;SET PROCESSOR PRIORITY TO 7.
                                MOV    @PRI07, R0
                                TRAP    C$SPRI
ENDINIT
                                L10012:
                                TRAP    C$INIT
TNUM == 0                    ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.

```

3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425

3426 012370
012370

3427
3434

3435 012370
012370
012370

104461

.SBTTL AUTODROP SECTION

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

BGNAUTO

ENDAUTO

L\$AUTO::

L10013: TRAP C\$AUTO

3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3468
3469
3470
3482
3483
3484
3485

012372
012372
012372 005767 167624
012376 001401
012400
012400 104433
012402
012402
012402 104432
012404 000002
012406
012406
012406 104412

.SBTTL CLEANUP CODING SECTION

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

BGNCLN

L\$CLEAN::

TST CTRLCF
BEQ 2\$
BRESET

;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 C\$RESET

2\$:

EXIT CLN

TRAP C\$EXIT
.WORD L10014 .

.EVEN

ENDCLN

L10014:
TRAP C\$CLEAN

```
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503 012410
      012410
3504
3505 012410
      012410 010046
      012412 012746 012434
      012416 012746 000002
      012422 010600
      012424 104417
      012426 062706 000006
3506 012432 000427
3507
3508 012434 045 101 040 DROP: .ASCIZ/*A UNIT#D6#A DROPPED FROM FURTHER TESTING.#N/
      012437 125 116 111
      012442 124 045 104
      012445 066 045 101
      012450 040 104 122
      012453 117 120 120
      012456 105 104 040
      012461 106 122 117
      012464 115 040 106
      012467 125 122 124
      012472 110 105 122
      012475 040 124 105
      012500 123 124 111
      012503 116 107 056
      012506 045 116 000
3509
3510 012512
3511
3512 012512
      012512 000167
      012514 000000
3513
3514
3515 012516
      012516
      012516 104453

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

BGNDU

L$DU::
PRINTF #DROP,RO ;REPORT UNIT THAT HAS BEEN DROPPED.
MOV RO,-(SP)
MOV #DROP,-(SP)
MOV #2,(SP)
MOV SP,RO
TRAP C$PNTF
ADD #6,SP

BR EDROP ;BRANCH AROUND THE MESSAGE.

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

EDROP: .EVEN
EXIT DU

FNDDU

L10015:
TRAP C$DU
```

```
3524
3525
3526          .SBTTL  ADD UNIT SECTION
3527
3528          ;**
3529          ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
3530          ; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
3531          ; TO THE TEST CYCLE.
3532          ;--
3533
3534 012520      BGNAU
3535 012520
3536          L$AU::
3537 012520      EXIT    AU
3538 012520      000167
3539 012522      000000
3540          .WORD    J$JMP
3541          .WORD    L10016-2-.
3542          .EVEN
3543          ENDAU
3544          L10016:
3545          TRAP      C$AU
```

```

3544 .SBTTL  HARDWARE TEST          - ADRA
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558 012526      BGNTST
      012526
3559      000001      TNUM == TNUM + 1      T1::
3560 012526 012767 000001 167472      MOV      @TNUM,TSTNUM      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
3561 012534 012767 177777 167460      MOV      @-1,CTRLCF      ;SET UP THE TEST NUMBER. (1)
3562 012542 012767 000145 171242      MOV      @101.,ERRNBR      ;INDICATE THAT WE ARE IN A TEST.
3563 012550 012767 004632 171236      MOV      @EM0103,ERRMSG      ;SET THE TEST ERROR NUMBER IN THE TABLE.
3564 012556 012767 006200 171232      MOV      @ER0101,ERRBLK      ;SET UP THE TEST FAILURE MESSAGE IN THE TABLF.
3565
3566
3567      ; SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
3568 012564 016767 165214 167456      MOV      4,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
3569 012572 012767 011530 165204      MOV      @TP4RTN,4      ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
3570 012600 005005      CLR      R5      ;CLEAR THE ERROR FLAGS.
3571
3572
3573
3574
3575      ; 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.
3576 012602 016700 167372      MOV      CSRA,R0      ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
3577 012606 012701 013000      MOV      @52,R1      ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
3578 012612 004767 175160      JSR      PC,CKTRAP      ;MOVE AND CHECK FOR TRAP.
3579 012616 103402      BCS      4,      ;IF NO TRAP, BYPASS ERROR.
3580 012620 052705 100001      BIS      @100001,R5      ;SET FATAL READ ERROR FLAGS.
3581 012624 042767 000017 000146 4:      BIC      @17,52,      ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
3582 012632 010100      MOV      R1,R0      ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
3583 012634 016701 167340      MOV      CSRA,R1      ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
3584 012640 004767 175132      JSR      PC,CKTRAP      ;MOVE AND CHECK FOR TRAP.
3585 012644 103403      BCS      6,      ;IF NO TRAP, BYPASS ERROR.
3586 012646 052705 100002      BIS      @100002,R5      ;SET FATAL WRITE ERROR FLAGS.
3587 012652 000434      BR      40,      ;EXIT AND REPORT FATAL ERROR.
3588
3589
3590      ; NOW, WE TEST EACH REGISTER FOR THIS LINE.
3591 012654 012702 000010      6:      MOV      @8.,R2      ;INIT REGISTER COUNTER TO 8.
3592 012660 016767 167314 000110      MOV      CSRA,50,      ;INITIALIZE THE REGISTER POINTER.
3593 012666 016700 000104      8:      MOV      50,R0      ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
3594 012672 012701 013000      MOV      @52,R1      ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
3595 012676 004767 175074      JSR      PC,CKTRAP      ;PERFORM THE MOVE, CHECK FOR TRAP.
3596 012702 103402      BCS      10,      ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
3597 012704 052705 100001      BIS      @100001,R5      ;SET FATAL READ ERROR FLAGS.
3598 012710 010100      10:      MOV      R1,R0      ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
3599 012712 016701 000060      MOV      50,R1      ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.

```

```

3600 012716 004767 175054      JSR      PC,CKTRAP      ;PERFORM THE MOVE, CHECK FOR TRAP.
3601 012722 103402              BCS      12$      ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
3602 012724 052705 100002      BIS      @100002,R5      ;SET FATAL WRITE ERROR FLAGS.
3603 012730 005267 000042      12$:      INC      50$      ;INCREMENT THE REGISTER
3604 012734 005267 000036              INC      50$      ; POINTER BY 2.
3605 012740 005302              DEC      R2      ;COUNT THE REGISTER.
3606 012742 001351              BNE      8$      ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
3607
3608
3609      ;*
3610      ; DONE CHECKING DEVICE REGISTER ADDRESSES.
3611      ; REPORT ANY ERRORS AND EXIT.
3612      ;*
3613 012744 016767 167300 165032 40$:      MOV      TP4VEC,4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
3614 012752 005705              TST      R5      ;CHECK THE ERROR FLAGS.
3615 012754 100012              BPL      60$      ;EXIT ROUTINE IF NO ERRORS.
3616
3617      ;*
3618      ; REPORT "DEVICE REGISTER ACCESS TEST FAILED"
3619      ;*
3619 012756              ERROR
3620 012756 104460
3621
3622              DODU      UNITN      ;DROP THIS UNIT FROM FUTHER TESTING.
3623 012760 016700 167212              MOV      UNITN,R0
3624 012764 104451              TRAP      C$DODU
3625 012766 005067 167230              CLR      CTRLCF      ;INDICATE NO CTRL-C ABORT FROM TEST.
3626 012772 104444              DOCLN      ;ABORT THIS SUB PASS.
3627 012774 000402              BR      60$      TRAP      C$DCLN
3628
3629      ;***** LOCAL STORAGE. *****
3630 012776 000000 50$:      .WORD      0      ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
3631 013000 000000 52$:      .WORD      0      ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
3632      ;***** END *****
3633 013002 005067 167214 60$:      CLR      CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
3634 013006              ENDTST
3635 013006 104401              L10017:
3636              TRAP      C$ETST

```

```

3635 .SBTTL  HARDWARE TEST - DTRMCS
3636 ;*****
3637 ;*
3638 ;* - DATA TERMINAL READY MODEM CONTROL SIGNAL TEST
3639 ;*
3640 ;* THIS TEST VERIFIES THAT THE DTR MODEM CONTROL SIGNAL IS WORKING
3641 ;* CORRECTLY. IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR STAGGERED
3642 ;* LOOPBACK IS SPECIFIED. THIS TEST USES THE LOOPED BACK SIGNALS RI
3643 ;* AND DSR TO TEST THE DTR SIGNAL. THIS TEST IS PERFORMED ON ALL
3644 ;* ACTIVE LINES.
3645 ;*
3646 ;-----
3647 013010 BGNTST
013010
3648
3649 ;* T2::
3650 ; ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
3651 013010 032767 000002 167156 ;
3652 013016 001002 ; BIT #BIT1,LOPBCK ;CHECK TYPE OF LOOPBACK MODE SELECTED.
3653 013020 000167 000504 BNE 2$
3654 013024 012700 000240 2$: JMP 60$ ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
013024 104441 ;ALLOW LTC INTERRUPTS.
013030 000002 MOV #PRI05,R0
TRAP C$SPRI
3655 013032 012767 000002 167166 TNUM -- TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
3656 013040 012767 177777 167154 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (78)
3657 013046 012767 000001 170734 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
3658 013054 012767 017171 170730 MOV #1,ERRTYP ;SET ERROR TYPE IN ERROR TABLE.
3659 013062 012767 004753 170724 MOV #7801,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
3660 ; MOV #EM7801,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
3661 ;
3662 ; RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.
3663 ; CLEAR TX AND RX INTERRUPT ENABLE BITS.
3664 ; THIS SUBROUTINE REPORTS ERROR >>>> 7801 <<<<.
3665 ;
3666 013070 004767 174732 ;
3667 013074 103402 ; JSR PC,CLNRST ;RESET THE DUT.
3668 013076 000167 000426 BCS 4$
3669 JMP 60$ ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
3670 ;
3671 ; SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
3672 013102 004767 174334 ;
3673 4$: JSR PC,ASLNTL ;SET UP THE ASSOCIATED LINE TABLES.
3674 ;
3675 ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
3676 ; THIS LOOP CLEARS ALL THE DTRS AND THEN SETS THEM INDIVIDUALLY AND WAITS FOR
3677 ; A RESPONSE ON THE ASSOCIATED RI AND DSR SIGNALS.
3678 ; THIS LOOP WILL CLEAR THE TX,IE AND RX,IE BITS IF THEY ARE SET.
3679 013106 005003 ;
3680 013110 010300 6$: CLR R3 ;CLEAR THE LINE COUNTER.
3681 013112 006300 MOV R3,R0
3682 013114 036067 002310 167050 ASL R0
3683 013122 001471 BIT BITBL(R0),ACTLNS
3684 BEQ 12$ ;DON'T TEST IF NOT ACTIVE LINE.
3685 ;
3686 ; CLEAR ALL THE DUT LNCTRL REGISTERS DTR BITS.
3687 013124 005000 ;
3688 013126 012705 177777 CLR R0 ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
MOV #MAPLNS,R5 ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.

```

```

3689 013132 004767 176272      JSR    PC,WTWLNCR      ;CLEAR ALL THE DTR DTR BITS.
3690 013136 012704 000074      MOV    #60,R4
3691 013142 004767 174774      JSR    PC,DELAY        ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.
3692
3693      ;*
3694      ; CHECK THAT AT LEAST ONE OF ASSOCIATED DSR OR RI IS CLEAR AND RECORD STATES.
3695
3695 013146 116304 003750      MOV    TXRLNB(R3),R4    ;GET THE ASSOCIATED LINE NUMBER.
3696 013152 010477 167022      MOV    R4,DCSRA        ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
3697 013156 017705 167024      MOV    #FSLSA,R5      ;GET THE STATE OF THE ASSOCIATED DSR, RI BITS.
3698 013162 012700 120000      MOV    #BIT15:BIT13,R0
3699 013166 040500              BIC     R5,R0                ;CHECK FOR BOTH DSR AND RI SET.
3700 013170 001431              BEQ     10$                ;GO REPORT DTR IS BAD IF BOTH ARE SET.
3701
3702      ;*
3703      ; SET THE DTR FOR THE SELECTED LINE AND WAIT FOR EITHER DSR OR RI TO SET.
3704
3704 013172 010377 167002      MOV    R3,DCSRA        ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
3705 013176 052777 001000      BIS    #BIT9,DLNCTRA  ;SET THE SELECTED LINE DTR.
3706 013204 012701 150074      MOV    #150074,R1     ;SPECIFY TO WAIT UP TO 60 MS FOR RI TO SET.
3707 013210 032705 100000      BIT    #BIT15,R5      ;CHECK PREVIOUS STATE OF DSR BIT.
3708 013214 001002              BNE     8$                ;GO USE RI IF DSR BIT WAS NOT CLEAR.
3709 013216 012701 170074      MOV    #170074,R1     ;SPECIFY TO WAIT UP TO 60 MS FOR DSR SET.
3710 013222 016702 166760      MOV    FSLSA,R2        ;SPECIFY TO LOOK IN STAT REG FOR BIT TO SET.
3711 013226 010477 166746      MOV    R4,DCSRA        ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
3712 013232 004767 176116      JSR    PC,WAIBIS      ;WAIT UP TO 60 MS FOR SIGNAL TO GO SET.
3713 013236 103423              BCS     12$                ;SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.
3714 013240 017700 166742      MOV    #FSLSA,R0      ;GET THE STATUS REGISTER CONTENTS.
3715 013244 042700 057777      BIC     #57777,R0      ;REMOVE ALL BUT THE DSR AND RI BITS.
3716 013250 040500              BIC     R5,R0                ;TEST FOR SIGNAL ONCE CLEAR, BUT NOW SET.
3717 013252 001015              BNE     12$                ;GO LOOP IF SIGNAL HAS GONE FROM CLR TO SET.
3718 013254
3719 013254 012767 017172      ;REPORT DTR MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
3720 013262 012767 006662      MOV    #7802,ERRNBR    ;SELECT THE ERROR NUMBER.
3721 013270 012701 005015      MOV    #ER7801,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
3722 013274      MOV    #EM7802,R1    ;SELECT THE ERROR MESSAGE.
3723      ERROR
3724
3725      TRAP    C:ERROR
3726
3726 013276 032767 000100      ;*
3727 013304 001511      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
3728
3728 013306 005203      ;*
3729 013310 020327      ;
3730 013314 002675      ;
3731
3732      ;
3733      ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
3734      ; THIS LOOP SETS ALL THE DTRS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
3735      ; A RESPONSE ON THE ASSOCIATED RI AND DSR SIGNALS.
3736      ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
3737
3737 013316 005003      ;*
3738 013320 010300      ;
3739 013322 006300      ;
3740 013324 036067      ;
3741 013332 001472      ;
3742
3743      ;
3744      ; SET ALL THE DUT LNCTRL REGISTERS DTR BITS.

```

```

3745
3746 013334 012700 001000      ;
3747 013340 012705 177777      MOV    #BIT9,R0      ;SPECIFY THAT DTR BITS ARE TO BE SET.
3748 013344 004767 176060      MOV    #MAPLNS,R5     ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
3749 013350 012704 000074      JSR    PC,WTWLNLC    ;SET ALL THE OUT DTR BITS.
3750 013354 004767 174562      MOV    #60.,R4
3751                                JSR    PC,DELAY      ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.
3752
3753      ; CHECK THAT AT LEAST ONE OF ASSOCIATED DSR OR RI IS SET AND RECORD STATES.
3754 013360 116304 003750      ;
3755 013364 010477 166610      MOV    TXRLNB(R3),R4    ;GET THE ASSOCIATED LINE NUMBER.
3756 013370 017705 166612      MOV    R4,BCSRA      ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
3757 013374 010500            MOV    #FSLSA,R5     ;GET THE STATE OF THE ASSOCIATED DSR, RI BITS.
3758 013376 042700 057777      MOV    R5,R0
3759 013402 001431            BIC    #57777,R0      ;CHECK FOR BOTH DSR AND RI CLEAR.
3760                                BEQ    18$          ;GO REPORT DTR IS BAD IF BOTH ARE CLEAR.
3761
3762      ; CLEAR THE DTR FOR THE SELECTED LINE AND WAIT FOR EITHER DSR OR RI TO CLEAR.
3763 013404 010377 166570      ;
3764 013410 042777 001000      MOV    R3,BCSRA      ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
3765 013416 012701 150074      BIC    #BIT9,BLNCTRA    ;CLEAR THE SELECTED LINE DTR.
3766 013422 032705 100000      MOV    #150074,R1     ;SPECIFY TO WAIT UP TO 60 MS FOR RI TO CLEAR.
3767 013426 001402            BIT    #BIT15,R5      ;CHECK PREVIOUS STATE OF DSR BIT.
3768 013430 012701 170074      BEQ    16$          ;GO USE RI IF DSR BIT WAS NOT SET.
3769 013434 016702 166546      MOV    #170074,R1     ;SPECIFY TO WAIT UP TO 60 MS FOR DSR CLEAR.
3770 013440 010477 166534      MOV    FSLSA,R2      ;SPECIFY TO LOOK IN STAT REG FOR BIT TO CLR.
3771 013444 004767 175630      MOV    R4,BCSRA      ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
3772 013450 103423            JSR    PC,WAIBIC      ;WAIT UP TO 60 MS FOR SIGNAL TO GO CLEAR.
3773 013452 017700 166530      BCS    20$          ;SELECT NEXT LINE AND LOOP IF SIGNAL IS CLEAR.
3774 013456 042705 057777      MOV    #FSLSA,R0      ;GET THE STATUS REGISTER CONTENTS.
3775 013462 040005            BIC    #57777,R5
3776 013464 001015            BIC    R0,R5
3777 013466            BNE    20$          ;TEST FOR SIGNAL ONCE SET, BUT NOW CLEAR.
3778 013466 012767 017173      18$: ;REPORT DTR MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
3779 013474 012767 006662      MOV    #7803.,ERRNBR    ;SELECT THE ERROR NUMBER.
3780 013502 012701 005015      MOV    #ER7801,ERRBLK    ;SELECT THE ERROR PRINT ROUTINE.
3781 013506            MOV    #EM7802,R1      ;SELECT THE ERROR MESSAGE.
3782                                ERROR
3783
3784                                TRAP    C$ERROR
3785 013510 032767 000100      ;
3786 013516 001404 166444      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
3787
3788 013520 005203            BIT    #BIT06,OPTION    ;EXIT WITH TEST FAILURE MESSAGE IF
3789 013522 020327 000020      BEQ    60$          ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
3790 013526 002674            ;DURING THE SOFTWARE QUESTIONS.
3791 013530 005067 166466      20$: INC    R3          ;SELECT THE NEXT LINE NUMBER.
3792 013534 012700 000340      CMP    R3,#NUMLNS    ;TEST FOR ALL LINES DONE.
3793 013542 104441            BLT    14$          ;LOOP IF NOT ALL LINES DONE.
3794 013542            CLR    CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
3795 013542 104401            SETPRI    #PRI07      ;DISABLE ALL INTERRUPTS.
3796                                MOV    #PRI07,R0
3797                                TRAP    C$SPRI
3798                                L10020:
3799                                TRAP    C$ETST
3800                                ENDTST

```

```

3795 .SBTTL  HARDWARE TEST          - RTSMCS -
3796 ;*****
3797 ;
3798 ; - REQUEST TO SEND MODEM CONTROL SIGNAL TEST
3799 ;
3800 ; THIS TEST VERIFIES THAT THE RTS MODEM CONTROL SIGNAL IS WORKING
3801 ; CORRECTLY. IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR STAGGERED
3802 ; LOOPBACK IS SPECIFIED. THIS TEST USES THE LOOPED BACK SIGNALS CTS
3803 ; AND DCD TO TEST THE RTS SIGNAL. THIS TEST IS PERFORMED ON ALL
3804 ; ACTIVE LINES.
3805 ;*****
3806 013544 BGNTST
      013544
3807
3808 ;
3809 ; ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
3810 013544 032767 000002 166422
3811 013552 001002
3812 013554 000167 000504
3813 013560 012700 000240
      013560 104441
      013564 000003
3814 013566 012767 000003 166432
3815 013574 012767 177777 166420
3816 013602 012767 000001 170200
3817 013610 012767 017335 170174
3818 013616 012767 005035 170170
3819
3820 ;
3821 ; RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.
3822 ; CLEAR TX AND RX INTERRUPT ENABLE BITS.
3823 ; THIS SUBROUTINE REPORTS ERROR >>>> 7901 <<<<.
3824 ;
3825 013624 004767 174176
3826 013630 103402
3827 013632 000167 000426
3828
3829 ;
3830 ; SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
3831 013636 004767 173600
3832
3833 ;
3834 ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
3835 ; THIS LOOP CLEARS ALL THE RTSS AND THEN SETS THEM INDIVIDUALLY AND WAITS FOR
3836 ; A RESPONSE ON THE ASSOCIATED CTS AND DCD SIGNALS.
3837 ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
3838 013642 005003
3839 013644 010300
3840 013646 006300
3841 013650 036067 002310 166314
3842 013656 001471
3843
3844 ;
3845 ; CLEAR ALL THE DUT LNCTRL REGISTERS RTS BITS.
3846 013660 005000
3847 013662 012705 177777
3848 013666 004767 175536

```

```

- RTSMCS -
*****
- REQUEST TO SEND MODEM CONTROL SIGNAL TEST
;
; THIS TEST VERIFIES THAT THE RTS MODEM CONTROL SIGNAL IS WORKING
; CORRECTLY. IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR STAGGERED
; LOOPBACK IS SPECIFIED. THIS TEST USES THE LOOPED BACK SIGNALS CTS
; AND DCD TO TEST THE RTS SIGNAL. THIS TEST IS PERFORMED ON ALL
; ACTIVE LINES.
*****
BGNTST
T3::
; ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
;
; BIT 0BIT1,LOPBCK ;CHECK TYPE OF LOOPBACK MODE SELECTED.
; BNE 1$
; JMP 60$ ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
1$: SETPRI 0PRI05 ;ALLOW LTC INTERRUPTS.
;
; MOV 0PRI05,R0
; TRAP C0SPRI
TNUM = TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
MOV 0TNUM,TSTNUM ;SET UP THE TEST NUMBER. (79)
MOV 0-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
MOV 01,ERRTYP ;SET ERROR TYPE IN ERROR TABLE.
MOV 07901,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
MOV 0EM7901,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
;
; RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS.
; THIS SUBROUTINE REPORTS ERROR >>>> 7901 <<<<.
;
; JSR PC,CLNRST ;RESET THE DUT.
; BCS 3$
; JMP 60$ ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
;
; SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
;
; JSR PC,ASLNTL ;SET UP THE ASSOCIATED LINE TABLES.
;
; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
; THIS LOOP CLEARS ALL THE RTSS AND THEN SETS THEM INDIVIDUALLY AND WAITS FOR
; A RESPONSE ON THE ASSOCIATED CTS AND DCD SIGNALS.
; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
;
; CLR R3 ;CLEAR THE LINE COUNTER.
2$: MOV R3,R0
; ASL R0
; BIT BITTBL(R0),ACTLNS
; BEQ 8$ ;DON'T TEST IF NOT ACTIVE LINE.
;
; CLEAR ALL THE DUT LNCTRL REGISTERS RTS BITS.
;
; CLR R0 ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
; MOV 0MAPLNS,R5 ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
; JSR PC,WTWLNC ;CLEAR ALL THE DUT RTS BITS.

```

```

3849 013672 012704 000074      MOV    #60.,R4
3850 013676 004767 174240      JSR    PC,DELAY      ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.
3851                                     ;*
3852                                     ; CHECK THAT AT LEAST ONE OF ASSOCIATED DCD OR CTS IS CLEAR AND RECORD STATES.
3853                                     ;-
3854 013702 116304 003750      MOV    TXRLNB(R3),R4      ;GET THE ASSOCIATED LINE NUMBER.
3855 013706 010477 166266      MOV    R4,#CSRA      ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
3856 013712 017705 166270      MOV    #FSLSA,R5      ;GET THE STATE OF THE ASSOCIATED DCD, CTS BITS.
3857 013716 012700 014000      MOV    #BIT12:BIT11,R0
3858 013722 040500              BIC    R5,R0      ;CHECK FOR BOTH DCD AND CTS SET.
3859 013724 001431              BEQ    6$      ;GO REPORT RTS IS BAD IF BOTH ARE SET.
3860                                     ;*
3861                                     ; SET THE RTS FOR THE SELECTED LINE AND WAIT FOR EITHER DCD OR CTS TO SET.
3862                                     ;-
3863 013726 010377 166246      MOV    R3,#CSRA      ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
3864 013732 052777 010000 166250  BIS    #BIT12,#LNCTRA      ;SET THE SELECTED LINE RTS.
3865 013740 012701 130074      MOV    #130074,R1      ;SPECIFY TO WAIT UP TO 60 MS FOR CTS TO SET.
3866 013744 032705 010000      BIT    #BIT12,R5      ;CHECK PREVIOUS STATE OF DCD BIT.
3867 013750 001002              BNE    4$      ;GO USE CTS IF DCD BIT WAS NOT CLEAR.
3868 013752 012701 140074      MOV    #140074,R1      ;SPECIFY TO WAIT UP TO 60 MS FOR DCD SET.
3869 013756 016702 166224 4$:    MOV    #FSLSA,R2      ;SPECIFY TO LOOK IN STAT REG FOR BIT TO SET.
3870 013762 010477 166212      MOV    R4,#CSRA      ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
3871 013766 004767 175362      JSR    PC,WAIBIS      ;WAIT UP TO 60 MS FOR SIGNAL TO GO SET.
3872 013772 103423              BCS    8$      ;SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.
3873 013774 017700 166206      MOV    #FSLSA,R0      ;GET THE STATUS REGISTER CONTENTS.
3874 014000 042700 163777      BIC    #163777,R0      ;REMOVE ALL BUT THE DCD AND CTS BITS.
3875 014004 040500              BIC    R5,R0      ;TEST FOR SIGNAL ONCE CLEAR, BUT NOW SET.
3876 014006 001015              BNE    8$      ;GO LOOP IF SIGNAL HAS GONE FROM CLR TO SET.
3877 014010 6$:    ;REPORT RTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
3878 014010 012767 017336 167774  MOV    #7902.,ERRNBR      ;SELECT THE ERROR NUMBER.
3879 014016 012767 006662 167772  MOV    #ER7801,ERRBLK      ;SELECT THE ERROR PRINT ROUTINE.
3880 014024 012701 005077      MOV    #EM7902,R1      ;SELECT THE ERROR MESSAGE.
3881 014030              ERROR      ;
3882                                     ;>>>> ERROR <<<<<.
3883                                     ; TRAP    C$ERROR
3884                                     ;*
3885                                     ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
3886                                     ;-
3887 014032 032767 000100 166122  BIT    #BIT06,OPTION      ;EXIT WITH TEST FAILURE MESSAGE IF
3888 014040 001511              BEQ    60$      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
3889                                     ; DURING THE SOFTWARE QUESTIONS.
3890 014042 005203 8$:    INC    R3      ;SELECT THE NEXT LINE NUMBER.
3891 014044 020327 000020      CMP    R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
3892 014050 002675      BLT    2$      ;LOOP IF NOT ALL LINES DONE.
3893                                     ;*
3894                                     ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
3895                                     ; THIS LOOP SETS ALL THE RTSS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
3896                                     ; A RESPONSE ON THE ASSOCIATED CTS AND DCD SIGNALS.
3897                                     ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
3898                                     ;-
3899 014052 005003      CLR    R3      ;CLEAR THE LINE COUNTER.
3900 014054 010300 10$:    MOV    R3,R0
3901 014056 006300      ASL    R0
3902 014060 036067 002310 166104  BIT    BITTBL(R0),ACTLNS
3903 014066 001472      BEQ    16$      ;DON'T TEST IF NOT ACTIVE LINE.
3904                                     ;*
3905                                     ; SET ALL THE DUT LNCTRL REGISTERS RTS BITS.
3906                                     ;-

```

```

3905 014070 012700 010000      MOV    #BIT12,R0      ;SPECIFY THAT RTS BITS ARE TO BE SET.
3906 014074 012705 177777      MOV    #MAPLNS,R5     ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
3907 014100 004767 175324      JSR     PC,WTWLNLC    ;SET ALL THE OUT RTS BITS.
3908 014104 012704 000074      MOV    #60.,R4
3909 014110 004767 174026      JSR     PC,DELAY      ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.
3910
3911      ; CHECK THAT AT LEAST ONE OF ASSOCIATED DCD OR CTS IS SET AND RECORD STATES.
3912
3913 014114 116304 003750      MOV     TXRLNB(R3),R4    ;GET THE ASSOCIATED LINE NUMBER.
3914 014120 010477 166054      MOV     R4,@CSRA      ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
3915 014124 017705 166056      MOV     @FSLSA,R5     ;GET THE STATE OF THE ASSOCIATED DCD. CTS BITS.
3916 014130 010500
3917 014132 042700 163777      MOV     R5,R0
3918 014136 001431      BIC     #163777,R0    ;CHECK FOR BOTH DCD AND CTS CLEAR.
3919      BEQ     14$        ;GO REPORT RTS IS BAD IF BOTH ARE CLEAR.
3920
3921      ; CLEAR THE RTS FOR THE SELECTED LINE AND WAIT FOR EITHER DCD OR CTS TO CLEAR.
3922
3922 014140 010377 166034      MOV     R3,@CSRA      ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
3923 014144 042777 010000 166036      BIC     #BIT12,@LNCTRA ;CLEAR THE SELECTED LINE RTS.
3924 014152 012701 130074      MOV     #130074,R1    ;SPECIFY TO WAIT UP TO 60 MS FOR CTS TO CLEAR.
3925 014156 032705 010000      BIT     #BIT12,R5     ;CHECK PREVIOUS STATE OF DCD BIT.
3926 014162 001402      BEQ     12$        ;GO USE CTS IF DCD BIT WAS NOT SET.
3927 014164 012701 140074      MOV     #140074,R1    ;SPECIFY TO WAIT UP TO 60 MS FOR DCD CLEAR.
3928 014170 016702 166012 12$:  MOV     FSLSA,R2      ;SPECIFY TO LOOK IN STAT REG FOR BIT TO CLR.
3929 014174 010477 166000      MOV     R4,@CSRA      ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
3930 014200 004767 175074      JSR     PC,WAIBIC     ;WAIT UP TO 60 MS FOR SIGNAL TO GO CLEAR.
3931 014204 103423      BCS     16$        ;SELECT NEXT LINE AND LOOP IF SIGNAL IS CLEAR.
3932 014206 017700 165774      MOV     @FSLSA,R0     ;GET THE STATUS REGISTER CONTENTS.
3933 014212 042705 163777      BIC     #163777,R5
3934 014216 040005      BIC     R0,R5
3935 014220 001015      BNE     16$        ;TEST FOR SIGNAL ONCE SET, BUT NOW CLEAR.
3936 014222      ;REPORT RTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
3937 014222 012767 017337 167562 14$:  MOV     #7903.,ERRNBR ;GO LOOP IF SIGNAL HAS GONE FROM SET TO CLR.
3938 014230 012767 006662 167560      MOV     #ER7801,ERRBLK ;SELECT THE ERROR NUMBER.
3939 014236 012701 005077      MOV     #EM7902,R1    ;SELECT THE ERROR PRINT ROUTINE.
3940 014242      ERROR      ;SELECT THE ERROR MESSAGE.
3941      ; >>>>> ERROR <<<<<<
3942      TRAP     C$ERROR
3943
3943      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
3944
3944 014244 032767 000100 165710      BIT     #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
3945 014252 001404      BEQ     60$        ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
3946
3947 014254 005203 16$:  INC     R3          ;DURING THE SOFTWARE QUESTIONS.
3948 014256 020327 000020      CMP     R3,#NUMLNS   ;SELECT THE NEXT LINE NUMBER.
3949 014262 002674      BLT     10$        ;TEST FOR ALL LINES DONE.
3950 014264 005067 165732 60$:  CLR     CTRLCF        ;LOOP IF NOT ALL LINES DONE.
3951 014270      SETPRI  #PRI07 ;INDICATE THAT WE ARE NOT WITHIN A TEST.
3952 014270 012700 000340      MOV     #PRI07,R0    ;DISABLE ALL INTERRUPTS.
3953 014274 104441      TRAP     C$SPRI
3954 014276      ENDTST
3955 014276 104401      L10021: TRAP     C$E$T

```

3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966 014300
014300
3967
3968
3969
3970 014300 032767 000002 165666
3971 014306 001002
3972 014310 000167 000420
3973 014314
014314 012700 000240
014320 104441
3974 000004
3975 014322 012767 000004 165676
3976 014330 012767 177777 165664
3977 014336 012767 000001 167444
3978 014344 012767 017501 167440
3979 014352 012767 005117 167434
3980
3981
3982
3983
3984
3985 014360 004767 173442
3986 014364 103402
3987 014366 000167 000342
3988
3989
3990
3991 014372 004767 173044
3992
3993
3994
3995
3996
3997
3998 014376 005003
3999 014400 010300
4000 014402 006300
4001 014404 036067 002310 165560
4002 014412 001454
4003
4004
4005
4006 014414 005000
4007 014416 012705 177777

```
.SBTTL  HARDWARE TEST                                DSRMS
*****
; - DATA SET READY MODEM SIGNAL TEST
;
; THIS TEST VERIFIES THAT THE DSR MODEM STATUS SIGNAL IS WORKING
; CORRECTLY. IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR STAGGERED
; LOOPBACK IS SPECIFIED. THIS TEST USES THE LOOPED BACK DTR SIGNALS
; TO TEST THE DSR SIGNAL. THIS TEST IS PERFORMED ON ALL THE ACTIVE
; LINES.
;
*****

BGNTST

T4::
; *
; ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
;
; BIT 0BIT1,LOPBCK ;CHECK TYPE OF LOOPBACK MODE SELECTED.
; BNE 2$
; JMP 60$
; SETPRI 0PRI05 ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
; ALLOW LTC INTERRUPTS.
;
; TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
; MOV 0TNUM,ISTNUM ;SET UP THE TEST NUMBER. (80)
; MOV 0-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
; MOV 01,ERRTYP ;SET ERROR TYPE IN ERROR TABLE.
; MOV 08001,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
; MOV 0EM8001,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
;
; RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS.
; THIS SUBROUTINE REPORTS ERROR >>>> 8001 <<<<.
;
; JSR PC,CLNRST ;RESET THE DUT.
; BCS 4$
; JMP 60$ ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
;
; SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
;
; JSR PC,ASLNTL ;SET UP THE ASSOCIATED LINE TABLES.
;
; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
; THIS LOOP CLEARS ALL THE DTRS AND THEN SETS THEM INDIVIDUALLY AND WAITS FOR
; A RESPONSE ON THE ASSOCIATED DSR SIGNAL.
; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
;
; CLR R3 ;CLEAR THE LINE COUNTER.
; MOV R3,R0
; ASL R0
; BIT BITTBL(R0),ACTLNS
; BEQ 10$ ;DON'T TEST IF NOT ACTIVE LINE.
;
; CLEAR ALL THE DUT LNCTRL REGISTERS DTR BITS.
;
; CLR R0
; MOV 0MAPLNS,R5 ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
; ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
```

```

4008 014422 004767 175002      JSR    PC,WTWLNCR      ;CLEAR ALL THE DUT DTR BITS.
4009 014426 012704 000050      MOV    #40.,R4
4010 014432 004767 173504      JSR    PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4011
4012      ;*
4013      ; CHECK THAT THE SPECIFIED DSR IS CLEAR.
4014 014436 010377 165536      MOV    R3,BCSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
4015 014442 032777 100000 165536 BIT    #BIT15,BSLSA
4016 014450 001020              BNE     B#                ;GO REPORT DSR IS BAD IF BIT IS NOT CLEAR.
4017
4018      ;*
4019      ; SET THE DTR FOR THE ASSOCIATED LINE.
4020      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED
4021      ; IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
4022 014452 116304 003750      MOV    TXRLNB(R3),R4    ;GET THE ASSOCIATED LINE NUMBER.
4023 014456 010477 165516      MOV    R4,BCSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
4024 014462 052777 001000 165520 BIS    #BIT9,BLNCTRA    ;SET THE ASSOCIATED LINE DTR.
4025
4026      ;*
4027      ; CHECK THAT THE SELECTED LINE DSR IS ACTIVE.
4028 014470 010377 165504      MOV    R3,BCSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
4029 014474 012701 170050      MOV    #170050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
4030 014500 016702 165502      MOV    FLSA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
4031 014504 004767 174644      JSR    PC,WAIBIS      ;WAIT FOR DSR TO BECOME SET OR TIMEOUT.
4032 014510 103415              BCS     10$                ;SKIP ERROR REPORT IF SELECTED DSR IS SET.
4033
4034
4035 014512              8$:    ;REPORT DSR MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
4036 014512 012767 017502 167272 MOV    #8002.,ERRNBR    ;SELECT THE ERROR NUMBER.
4037 014520 012767 006662 167270 MOV    #ER7801,ERRBLK    ;SELECT THE ERROR PRINT ROUTINE.
4038 014526 012701 005163      MOV    #EM8002,R1      ;SELECT THE ERROR MESSAGE.
4039 014532
4040 014532 104460              ERROR
4041
4042
4043      ;*
4044      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4045 014534 032767 000100 165420 BIT    #BIT06,OPTION
4046 014542 001474              BEQ     60$
4047
4048      ;*
4049      ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
4050 014546 020327 000020      INC    R3                ;SELECT THE NEXT LINE NUMBER.
4051 014552 002712              CMP    R3,#NUMLNS          ;TEST FOR ALL LINES DONE.
4052
4053      ;*
4054      ; THIS LOOP SETS ALL THE DTRS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
4055      ; A RESPONSE ON THE SELECTED DSR SIGNAL.
4056      ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
4057 014554 005003              CLR    R3                ;CLEAR THE LINE COUNTER.
4058 014556 010300      12$:    MOV    R3,R0
4059 014560 006300              ASL    R0
4060 014562 036067 002310 165402 BIT    BITBL(R0),ACTLNS
4061 014570 001455              BEQ     16$                ;DON'T TEST IF NOT ACTIVE LINE.
4062
4063      ;*
4064      ; SET ALL THE DUT LNCTRL REGISTERS DTR BITS.

```

TRAP C\$ERROR

```

4064
4065 014572 012700 001000      ;
4066 014576 012705 177777      MOV    #BIT9,R0      ;SPECIFY THAT DTR BITS ARE TO BE SET.
4067 014602 004767 174622      MOV    #MAPLNS,R5     ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
4068 014606 012704 000050      JSR    PC,WTWLNLC     ;SET ALL THE OUT DTR BITS.
4069 014612 004767 173324      MOV    #40.,R4
4070                               JSR    PC,DELAY      ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4071
4072      ;+
4073 014616 010377 165356      ; CHECK THAT THE SPECIFIED DSR IS SET.
4074 014622 032777 100000 165356      ; -
4075 014630 001420      MOV    R3,#CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
4076                               BIT    #BIT15,#FSLSA
4077                               BEQ    14$      ;GO REPORT DSR IS BAD IF BIT IS NOT SET.
4078      ;+
4079      ; CLEAR THE DTR FOR THE ASSOCIATED LINE.
4080      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED
4081      ; IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
4082      ; -
4083 014632 116304 003750      MOV    TXRLNB(R3),R4      ;GET THE ASSOCIATED LINE NUMBER.
4084 014636 010477 165336      MOV    R4,#CSRA      ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
4085 014642 042777 001000 165340      BIC    #BIT9,#LNCTRA      ;CLEAR THE ASSOCIATED LINE DTR.
4086      ;+
4087      ; CHECK THAT THE SELECTED LINE DSR IS CLEAR.
4088      ; -
4089 014650 010377 165324      MOV    R3,#CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
4090 014654 012701 170050      MOV    #170050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
4091 014660 016702 165322      MOV    FSLSA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
4092 014664 004767 174410      JSR    PC,WAIBIC      ;WAIT FOR DSR TO BECOME CLEAR OR TIMEOUT.
4093 014670 103415      BCS    16$      ;SKIP ERROR REPORT IF SELECTED DSR IS CLEAR.
4094 014672      14$:      ;REPORT DSR MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
4095 014672 012767 017503 167112      MOV    #8003.,ERRNBR      ;SELECT THE ERROR NUMBER.
4096 014700 012767 006662 167110      MOV    #ER7801,ERRBLK      ;SELECT THE ERROR PRINT ROUTINE.
4097 014706 012701 005163      MOV    #EM8002,R1      ;SELECT THE ERROR MESSAGE.
4098 014712 104460      ERROR
4099
4100      TRAP    C$ERROR
4101
4102      ;+
4103      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4104      ; -
4105 014714 032767 000100 165240      BIT    #BIT06,OPTION
4106 014722 001404      BEQ    60$      ;EXIT WITH TEST FAILURE MESSAGE IF
4107                               ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4108                               ;DURING THE SOFTWARE QUESTIONS.
4109      16$:      INC    R3      ;SELECT THE NEXT LINE NUMBER.
4110      CMP    R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
4111      BLT    12$      ;LOOP IF NOT ALL LINES DONE.
4112      60$:      CLR    CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
4113      SETPRI #PRI07      ;DISABLE ALL INTERRUPTS.
4114
4115      MOV    #PRI07,R0
4116      TRAP    C$SPRI
4117
4118      ENDTST
4119
4120      L10022:
4121      TRAP    C$ETST
4122 014746 104401

```

```

4115 .SBTTL HARDWARE TEST - RINGI
4116 *****
4117 ;* - RING INDICATOR MODEM SIGNAL TEST
4118 ;*
4119 ;* THIS TEST VERIFIES THAT THE RI MODEM STATUS SIGNAL IS WORKING
4120 ;* CORRECTLY. IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR STAGGERED
4121 ;* LOOPBACK IS SPECIFIED. THIS TEST USES THE LOOPEO BACK DTR SIGNALS
4122 ;* TO TEST THE RI SIGNAL. THIS TEST IS PERFORMED ON ALL THE ACTIVE
4123 ;* LINES.
4124 ;*
4125 ;* -*****
4126
4127 014750 BGNTST
4128 014750
4129
4130 ;* TS::
4131 ; ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
4132 ;*
4133 BIT #BIT1,LOPBECK ;CHECK TYPE OF LOOPBACK MODE SELECTED.
4134 BNE 2$
4135 JMP 60$ ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
4136 2$: SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
4137
4138 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
4139 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (81)
4140 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
4141 MOV #1,ERRTYP ;SET ERROR TYPE IN ERROR TABLE.
4142 MOV #8101,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
4143 MOV #EM8101,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
4144
4145 ;*
4146 ; RESET THE DUT TO A KNOWN STATE. REMOVE STATUS CODES FROM THE FIFO.
4147 ; CLEAR TX AND RX INTERRUPT ENABLE BITS.
4148 ; THIS SUBROUTINE REPORTS ERROR >>>> 8101 <<<<.
4149 ;*
4150 JSR PC,CLNRST ;RESET THE DUT.
4151 BCS 4$
4152 JMP 60$ ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
4153
4154 ;*
4155 ; SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
4156 ;*
4157 4$: JSR PC,ASLNTL ;SET UP THE ASSOCIATED LINE TABLES.
4158
4159 ;*
4160 ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
4161 ; THIS LOOP CLEARS ALL THE DTRS AND THEN SETS THEM INDIVIDUALLY AND WAITS FOR
4162 ; A RESPONSE ON THE ASSOCIATED RI SIGNAL.
4163 ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
4164 ;*
4165 6$: CLR R3 ;CLEAR THE LINE COUNTER.
4166 MOV R3,R0
4167 ASL R0
4168 BIT BITBL(R0),ACTLNS
4169 BEQ 10$ ;DON'T TEST IF NOT ACTIVE LINE.
4170
4171 ;*
4172 ; CLEAR ALL THE DUT LNCTRL REGISTERS DTR BITS.
4173 ;*
4174 10$: CLR R0 ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
4175 MOV #MAPLNS,R5 ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.

```

```
4169 015072 004767 174332      JSR    PC,WTWLN      ;CLEAR ALL THE DUT DTR BITS.
4170 015076 012704 000050      MOV    #40.,R4
4171 015102 004767 173034      JSR    PC,DELAY      ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4172
4173      ;+
4174      ; CHECK THAT THE SPECIFIED RI IS CLEAR.
4175      ;-
4175 015106 010377 165066      MOV    R3,@CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
4176 015112 032777 020000 165066  BI1    @BIT13,@FSLSA
4177 015120 001020              BNE     B#              ;GO REPORT RI IS BAD IF BIT IS NOT CLEAR.
4178
4179      ;+
4180      ; SET THE DTR FOR THE ASSOCIATED LINE.
4181      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED
4182      ; IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
4183      ;-
4183 015122 116304 003750      MOV8   TXRLNB(R3),R4  ;GET THE ASSOCIATED LINE NUMBER.
4184 015126 010477 165046      MOV    R4,@CSRA      ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
4185 015132 052777 001000 165050  BIS    @BIT9,@LNCTRA  ;SET THE ASSOCIATED LINE DTR.
4186
4187      ;+
4188      ; CHECK THAT THE SELECTED LINE RI IS ACTIVE.
4189      ;-
4189 015140 010377 165034      MOV    R3,@CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
4190 015144 012701 150050      MOV    #150050,R1    ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
4191 015150 016702 165032      MOV    FLSA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
4192 015154 004767 174174      JSR    PC,WAIBIS     ;WAIT FOR RI TO BECOME SET OR TIMEOUT.
4193 015160 103415              BCS     10#              ;SKIP ERROR REPORT IF SELECTED RI IS SET.
4194
4195
4196 015162              8#:      ;REPORT RI MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
4197 015162 012767 017646 166622  MOV    @8102.,ERRNBR  ;SELECT THE ERROR NUMBER.
4198 015170 012767 006662 166620  MOV    @ER7801,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
4199 015176 012701 005272              MOV    @EM8102,R1    ;SELECT THE ERROR MESSAGE.
4200 015202              ERROR
4201              TRAP      C#ERROR
4202
4203      ;+
4204      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4205      ;-
4205 015204 032767 000100 164750  BIT    @BIT06,OPTION  ;EXIT WITH TEST FAILURE MESSAGE IF
4206 015212 001474              BEQ     60#              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4207
4208
4209      ;+
4210      ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
4211      ; THIS LOOP SETS ALL THE DTRS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
4212      ; A RESPONSE ON THE SELECTED RI SIGNAL.
4213      ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
4214      ;-
4214 015214 005203              10#:  INC    R3              ;SELECT THE NEXT LINE NUMBER.
4215 015216 020327 000020      CMP    R3,@NUMLNS      ;TEST FOR ALL LINES DONE.
4216 015222 002712              BLT     6#              ;LOOP IF NOT ALL LINES DONE.
4217
4218      ;+
4219      ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
4220      ; THIS LOOP SETS ALL THE DTRS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
4221      ; A RESPONSE ON THE SELECTED RI SIGNAL.
4222      ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
4223      ;-
4223 015224 005003              12#:  CLR    R3              ;CLEAR THE LINE COUNTER.
4224 015226 010300      MOV    R3,R0
4225 015230 006300      ASL    R0
4226 015232 036067 002310 164732  BIT    BITBL(R0),ACTLNS
4227 015240 001455              BEQ     16#              ;DON'T TEST IF NOT ACTIVE LINE.
4228
4229      ;+
4230      ; SET ALL THE DUT LNCTRL REGISTERS DTR BITS.
```

```

4225
4226 015242 012700 001000
4227 015246 012705 177777
4228 015252 004767 174152
4229 015256 012704 000050
4230 015262 004767 172654
4231
4232
4233
4234 015266 010377 164706
4235 015272 032777 020000 164706
4236 015300 001420
4237
4238
4239
4240
4241
4242 015302 116304 003750
4243 015306 010477 164666
4244 015312 042777 001000 164670
4245
4246
4247
4248 015320 010377 164654
4249 015324 012701 150050
4250 015330 016702 164652
4251 015334 004767 173740
4252 015340 103415
4253
4254 015342
4255 015342 012767 017647 166442
4256 015350 012767 006662 166440
4257 015356 012701 005272
4258 015362
      015362 104460
4259
4260
4261
4262
4263 015364 032767 000100 164570
4264 015372 001404
4265
4266
4267 015374 005203
4268 015376 020327 000020
4269 015402 002711
4270
4271 015404 005067 164612
4272 015410
      015410 012700 000340
      015414 104441
4273
4274 015416
      015416
      015416 104401

      MOV      #BIT9,R0      ;SPECIFY THAT DTR BITS ARE TO BE SET.
      MOV      #MAPLNS,R5    ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
      JSR      PC,WTW.NC      ;SET ALL THE DUT DTR BITS.
      MOV      #40.,R4
      JSR      PC,DELAY      ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.

      ; CHECK THAT THE SPECIFIED RI IS SET.
      MOV      R3,BCSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
      BIT      #BIT13,BSLSA
      BEQ      14:          ;GO REPORT RI IS BAD IF BIT IS NOT SET.

      ; CLEAR THE DTR FOR THE ASSOCIATED LINE.
      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED
      ; IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
      MOVB     TXRLNB(R3),R4  ;GET THE ASSOCIATED LINE NUMBER.
      MOV      R4,BCSRA      ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
      BIC      #BIT9,SLNCTRA ;CLEAR THE ASSOCIATED LINE DTR.

      ; CHECK THAT THE SELECTED LINE RI IS CLEAR.
      MOV      R3,BCSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
      MOV      #150050,R1    ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
      MOV      FLSA,R2       ;PASS THE ADDRESS OF THE REGISTER TO TEST.
      JSR      PC,WAIBIC     ;WAIT FOR RI TO BECOME CLEAR OR TIMEOUT.
      BCS      16:          ;SKIP ERROR REPORT IF SELECTED RI IS CLEAR.

14:      ;REPORT RI MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
      MOV      #B103,ERRNBR  ;SELECT THE ERROR NUMBER.
      MOV      #ER7801,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
      MOV      #EM8102,R1    ;SELECT THE ERROR MESSAGE.
      ERROR

                                TRAP      C:ERROR

      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
      BIT      #BIT06,OPTION
      BEQ      60:

16:      INC      R3
      CMP      R3,#NUMLNS
      BLT      12:          ;SELECT THE NEXT LINE NUMBER.
                                ;TEST FOR ALL LINES DONE.
                                ;LOOP IF NOT ALL LINES DONE.

60:      CLR      CTRLCF
      SETPRI   #PRI07
                                ;INDICATE THAT WE ARE NOT WITHIN A TEST.
                                ;DISABLE ALL INTERRUPTS.

                                MOV      #PRI07,R0
                                TRAP      C:SPRI

      ENDTST

                                L10023:
                                TRAP      C:ETST

```

```

.SBTL  HARDWARE TEST          - CTSMS
*****
;
;   - CLEAR TO SEND MODEM SIGNAL TEST
;
;   THIS TEST VERIFIES THAT THE CTS MODEM STATUS SIGNAL IS WORKING
;   CORRECTLY.  IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR STAGGERED
;   LOOPBACK IS SPECIFIED.  THIS TEST USES THE LOOPED BACK RTS SIGNALS
;   TO TEST THE CTS SIGNAL.  THIS TEST IS PERFORMED ON ALL THE ACTIVE
;   LINES.
;
*****

BGNTST

;
;   T6::
;   ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
;
;   BIT      0BIT1,LOPBCK      ;CHECK TYPE OF LOOPBACK MODE SELECTED.
;   BNE      2$
;   JMP      60$               ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
2$: SETPRI   0PRI05            ;ALLOW LTC INTERRUPTS.

;
;   MOV      0PRI05,R0
;   TRAP     C$SPRI
;
;   TNUM = TNUM + 1            ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
;   MOV      0TNUM,TSTNUM      ;SET UP THE TEST NUMBER. (82)
;   MOV      0-1,CTRLCF        ;INDICATE THAT WE ARE IN A TEST.
;   MOV      01,ERRTP          ;SET ERROR TYPE IN ERROR TABLE.
;   MOV      08201,ERRNBR      ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
;   MOV      0EM8201,ERRMSG     ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
;
;   RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.
;   CLEAR TX AND RX INTERRUPT ENABLE BITS.
;   THIS SUBROUTINE REPORTS ERROR >>>> 8201 <<<<.
;
;   JSR      PC,CLNRST         ;RESET THE DUT.
;   BCS      4$
;   JMP      60$               ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
;
;   SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
;
4$: JSR      PC,ASLNTL         ;SET UP THE ASSOCIATED LINE TABLES.
;
;   SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
;   THIS LOOP CLEARS ALL THE RTS'S AND THEN SETS THEM INDIVIDUALLY AND WAITS FOR
;   A RESPONSE ON THE ASSOCIATED CTS SIGNAL.
;   THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
;
;   CLR      R3                ;CLEAR THE LINE COUNTER.
6$: MOV      R3,R0
;   ASL      R0
;   BIT      BITTBL(R0),ACTLNS
;   BEQ      10$               ;DON'T TEST IF NOT ACTIVE LINE.
;
;   CLEAR ALL THE DUT LNCTRL REGISTERS RTS BITS.
;
;   CLR      R0                ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
;   MOV      0MAPLNS,R5        ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.

```

```

4330 015542 004767 173662      JSR      PC,WTWLCN      ;CLEAR ALL THE DUT RTS BITS.
4331 015546 012704 000050      MOV      #40,R4
4332 015552 004767 172364      JSR      PC,DELAY      ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4333
4334      ; CHECK THAT THE SPECIFIED CTS IS CLEAR.
4335
4336 015556 010377 164416      MOV      R3,DCSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
4337 015562 032777 004000 164416 BIT      #BIT11,DFSLSA
4338 015570 001020      BNE      B#      ;GO REPORT CTS IS BAD IF BIT IS NOT CLEAR.
4339
4340      ; SET THE RTS FOR THE ASSOCIATED LINE.
4341      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, RTS WILL NOT HAVE BEEN TESTED
4342      ; IN THE RTS TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
4343
4344 015572 116304 003750      MOV      TXRLNB(R3),R4 ;GET THE ASSOCIATED LINE NUMBER.
4345 015576 010477 164376      MOV      R4,DCSRA      ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
4346 015602 052777 010000 164400 BIS      #BIT12,DLNCTRA ;SET THE ASSOCIATED LINE RTS.
4347
4348      ; CHECK THAT THE SELECTED LINE CTS IS ACTIVE.
4349
4350 015610 010377 164364      MOV      R3,DCSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
4351 015614 012701 130050      MOV      #130050,R1    ;PASS TIMEOUT OF 40 MILLI SEC. AND BIT TO TEST.
4352 015620 016702 164362      MOV      FSLSA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
4353 015624 004767 173524      JSR      PC,WAIBIS      ;WAIT FOR CTS TO BECOME SET OR TIMEOUT.
4354 015630 103415      BCS      10$          ;SKIP ERROR REPORT IF SELECTED CTS IS SET.
4355
4356
4357 015632      B#      ;REPORT CTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
4358 015632 012767 020012 166152 MOV      #8202,ERRNBR ;SELECT THE ERROR NUMBER.
4359 015640 012767 006662 166150 MOV      #ER7801,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
4360 015646 012701 005401      MOV      #EM8202,R1    ;SELECT THE ERROR MESSAGE.
4361 015652      ERROR
4362      ;
4363      ;
4364      ;
4365      ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4366 015654 032767 000100 164300 BIT      #BIT06,OPTION
4367 015662 001474      BEQ      60$          ;EXIT WITH TEST FAILURE MESSAGE IF
4368      ; NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4369      ; DURING THE SOFTWARE QUESTIONS.
4370 015664 005203      10$: INC      R3          ;SELECT THE NEXT LINE NUMBER.
4371 015666 020327 000020      CMP      R3,#NUMLNS ;TEST FOR ALL LINES DONE.
4372 015672 002712      BLT      6$          ;LOOP IF NOT ALL LINES DONE.
4373
4374      ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
4375      ; THIS LOOP SETS ALL THE RTSS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
4376      ; A RESPONSE ON THE SELECTED CTS SIGNAL
4377      ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
4378
4379 015674 005003      ;
4380 015676 010300      CLR      R3          ;CLEAR THE LINE COUNTER.
4381 015700 006300      12$: MOV      R3,R0
4382 015702 036067 002310 164262 ASL      R0
4383 015710 001455      BIT      BITTBL(R0),ACTLNS
4384      BEQ      16$          ;DON'T TEST IF NOT ACTIVE LINE.
4385
      ; SET ALL THE DUT LNCTRL REGISTERS RTS BITS.

```

Address	Offset	Value	Label	Instruction	Comment
4386	015712	012700	010000		
4388	015716	012705	177777		
4389	015722	004767	173502		
4390	015726	012704	000050		
4391	015732	004767	172204		
4392					
4393					
4394					
4395	015736	010377	164236		
4396	015742	032777	004000	164236	
4397	015750	001420			
4398					
4399					
4400					
4401					
4402					
4403	015752	116304	003750		
4404	015756	010477	164216		
4405	015762	042777	010000	164220	
4406					
4407					
4408					
4409	015770	010377	164204		
4410	015774	012701	130050		
4411	016000	016702	164202		
4412	016004	004767	173270		
4413	016010	103415			
4414					
4415	016012				
4416	016012	012767	020013	165772	
4417	016020	012767	006662	165770	
4418	016026	012701	005401		
4419	016032				
4420	016032	104460			
4421					
4422					
4423					
4424	016034	032767	000100	164120	
4425	016042	001404			
4426					
4427					
4428	016044	005203			
4429	016046	020327	000020		
4430	016052	002711			
4431					
4432	016054	005067	164142		
4433	016060				
4434	016060	012700	000340		
4435	016064	104441			
4436					
4437					
4438					
4439					
4440					
4441					
4442					
4443					
4444					
4445					
4446					
4447					
4448					
4449					
4450					
4451					
4452					
4453					
4454					
4455					
4456					
4457					
4458					
4459					
4460					
4461					
4462					
4463					
4464					
4465					
4466					
4467					
4468					
4469					
4470					
4471					
4472					
4473					
4474					
4475					
4476					
4477					
4478					
4479					
4480					
4481					
4482					
4483					
4484					
4485					

```

4437 .SBTTL HARDWARE TEST - DCDMS
4438 ;*****
4439 ;*
4440 ;* - DATA CARRIER DETECTED MODEM SIGNAL TEST
4441 ;*
4442 ;* THIS TEST VERIFIES THAT THE DCD MODEM STATUS SIGNAL IS WORKING
4443 ;* CORRECTLY. IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR STAGGERED
4444 ;* LOOPBACK IS SPECIFIED. THIS TEST USES THE LOOPED BACK RTS SIGNALS
4445 ;* TO TEST THE DCD SIGNAL. THIS TEST IS PERFORMED ON ALL THE ACTIVE
4446 ;* LINES.
4447 ;*
4448 ;*****
4449 016070 BGNTST
4450 016070 T7::
4451 ;*
4452 ;* ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
4453 016070 032767 000002 164076 ;
4454 016076 001002 ; BIT #BIT1,LOPCK ;CHECK TYPE OF LOOPBACK MODE SELECTED.
4455 016100 000167 000420 ; BNE 2$
4456 016104 012700 000240 ; JMP 60$ ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
4457 016110 104441 ; SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
4458 016112 000007 ; MOV #PRI05,R0
4459 016120 012767 000007 164106 ; TRAP C$SPRI
4460 016126 012767 177777 164074 ; TNUM = TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
4461 016134 012767 000001 165654 ; MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (83)
4462 016142 012767 005445 165644 ; MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
4463 ; MOV #1,ERRTP ;SET ERROR TYPE IN ERROR TABLE.
4464 ; MOV #8301,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
4465 ; MOV #EM8301,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
4466 ;*
4467 ;* RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.
4468 016150 004767 171652 ; CLEAR TX AND RX INTERRUPT ENABLE BITS.
4469 016154 103402 ; THIS SUBROUTINE REPORTS ERROR >>>> 8301 <<<<.
4470 016156 000167 000342 ;*
4471 ;* JSR PC,CLNRST ;RESET THE DUT.
4472 ;* BCS 4$
4473 ;* JMP 60$ ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
4474 016162 004767 171254 ;*
4475 ;* SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
4476 ;* JSR PC,ASLNTL ;SET UP THE ASSOCIATED LINE TABLES.
4477 ;*
4478 ;* SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
4479 ;* THIS LOOP CLEARS ALL THE RTSS AND THEN SETS THEM INDIVIDUALLY AND WAITS FOR
4480 ;* A RESPONSE ON THE ASSOCIATED DCD SIGNAL.
4481 016166 005003 ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
4482 016170 010300 ;*
4483 016172 006300 ;* CLR R3 ;CLEAR THE LINE COUNTER.
4484 016174 036067 002310 163770 6$: MOV R3,R0
4485 016202 001454 ; ASL R0
4486 ; BIT BITBL(R0),ACTLNS
4487 ; BEQ 10$ ;DON'T TEST IF NOT ACTIVE LINE.
4488 ;*
4489 016204 005000 ; CLEAR ALL THE DUT LNCTRL REGISTERS RTS BITS.
4490 016206 012705 177777 ;*
4491 ;* CLR R0 ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
4492 ;* MOV #MAPLNS,R5 ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.

```

```

4491 016212 004767 173212      JSR    PC,WTLNC      ;CLEAR ALL THE OUT RTS BITS.
4492 016216 012704 000050      MOV    #40.,R4
4493 016222 004767 171714      JSR    PC,DELAY      ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4494
4495      ;*
4496      ; CHECK THAT THE SPECIFIED DCD IS CLEAR.
4497 016226 010377 163746      MOV    R3,@CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
4498 016232 032777 010000 163746 BIT    @BIT12,@FSLSA
4499 016240 001020              BNE     8$              ;GO REPORT DCD IS BAD IF BIT IS NOT CLEAR.
4500
4501      ;*
4502      ; SET THE RTS FOR THE ASSOCIATED LINE.
4503      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, RTS WILL NOT HAVE BEEN TESTED
4504      ; IN THE RTS TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
4505 016242 116304 003750      MOV    TXRLNB(R3),R4 ;GET THE ASSOCIATED LINE NUMBER.
4506 016246 010477 163726      MOV    R4,@CSRA      ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
4507 016252 052777 010000 163730 BIS    @BIT12,@LNCTRA ;SET THE ASSOCIATED LINE RTS.
4508
4509      ;*
4510      ; CHECK THAT THE SELECTED LINE DCD IS ACTIVE.
4511 016260 010377 163714      MOV    R3,@CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
4512 016264 012701 140050      MOV    #140050,R1    ;PASS TIMEOUT OF 40 MILLI-SEC. AND BIT TO TEST.
4513 016270 016702 163712      MOV    FSLSA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
4514 016274 004767 173054      JSR    PC,WAIBIS     ;WAIT FOR DCD TO BECOME SET OR TIMEOUT.
4515 016300 103415              BCS     10$            ;SKIP ERROR REPORT IF SELECTED DCD IS SET.
4516
4517
4518 016302              8$:      ;REPORT DCD MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
4519 016302 012767 020156 165502 MOV    #8302.,ERRNBR ;SELECT THE ERROR NUMBER.
4520 016310 012767 006662 165500 MOV    #ER7801,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
4521 016316 012701 005511      MOV    #EM8302,R1    ;SELECT THE ERROR MESSAGE.
4522 016322
4523 016322 104460              ERROR
4524
4525
4526      ;*
4527 016324 032767 000100 163630 ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4528 016332 001474              BIT    @BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
4529
4530              BEQ    60$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4531
4532              ;DURING THE SOFTWARE QUESTIONS.
4533 016334 005203 10$:      INC    R3 ;SELECT THE NEXT LINE NUMBER.
4534 016336 020327 000020      CMP    R3,#NUMLNS ;TEST FOR ALL LINES DONE.
4535 016342 002712              BLT    6$ ;LOOP IF NOT ALL LINES DONE.
4536
4537      ;*
4538      ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
4539      ; THIS LOOP SETS ALL THE RTSS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
4540      ; A RESPONSE ON THE SELECTED DCD SIGNAL.
4541      ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
4542 016344 005003 12$:      CLR    R3 ;CLEAR THE LINE COUNTER.
4543 016346 010300      MOV    R3,R0
4544 016350 006300      ASL    R0
4545 016352 036067 002310 163612 BIT    BITTBL(R0),ACTLNS
4546 016360 001455      BEQ    16$ ;DON'T TEST IF NOT ACTIVE LINE.
4547
4548      ;*
4549      ; SET ALL THE OUT LNCTRL REGISTERS RTS BITS.

```

TRAP C\$ERROR


```
4598 .SBTTL HARDWARE TEST DTRINT
4599 :*****
4600 : DATA TERMINAL READY SIGNAL INTERACTIONS TEST
4601 :
4602 : THIS TEST VERIFIES THAT THE DTR SIGNAL (AND THE LOOPED BACK DSR AND
4603 : RI STATUS SIGNALS) DO NOT INTERACT WITH ANY OTHER MODEM STATUS SIGNALS.
4604 : IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR STAGGERED LOOPBACK IS
4605 : SPECIFIED. THIS TEST IS PERFORMED ON ALL ACTIVE LINES.
4606 :
4607 : *****
4608 :
4609 016540 BGNTST
016540
4610 TB::
4611 :
4612 : ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
4613 016540 032767 000002 163426
4614 016546 001002
4615 016550 000167 000400
4616 016554 012700 000240
016554 104441
016560 000010
4617 016562 012767 000010 163436
4618 016570 012767 177777 163424
4619 016576 012767 000001 165204
4620 016604 012767 020321 165200
4621 016612 012767 005555 165174
4622
4623
4624
4625
4626
4627
4628 016620 004767 171202
4629 016624 103402
4630 016626 000167 000322
4631
4632
4633
4634 016632 004767 170604
4635
4636
4637
4638
4639
4640
4641 016636 005003
4642 016640 010300
4643 016642 006300
4644 016644 036067 002310 163320
4645 016652 001450
4646
4647
4648
4649 016654 005000
4650 016656 012705 177777
4651 016662 004767 172542

;*****
;
; BIT #BIT1,LOPBCK ;CHECK TYPE OF LOOPBACK MODE SELECTED.
; BNE 2$
; JMP 60$ ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
2$: SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
;
; MOV #PRI05,R0
; TPAP C$SPRI
;
; TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
; MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (84)
; MOV #1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
; MOV #1,ERRTYP ;SET ERROR TYPE IN ERROR TABLE.
; MOV #8401,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
; MOV #EM8401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
;
; RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS.
; THIS SUBROUTINE REPORTS ERROR >>>> 8401 <<<<.
;
; JSR PC,CLNRST ;RESET THE DUT.
; BCS 4$
; JMP 60$ ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
;
; SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
;
; JSR PC,ASLNTL ;SET UP THE ASSOCIATED LINE TABLES.
;
; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
; THIS LOOP CLEARS ALL THE DTRS AND THEN SETS THEM INDIVIDUALLY AND CHECKS
; FOR ANY RESPONSES ON SIGNALS OTHER THAN THE ASSOCIATED RI AND DSR SIGNALS.
; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
;
; CLR R3 ;CLEAR THE LINE COUNTER.
6$: MOV R3,R0
ASL R0
BIT BITTBL(R0),ACTLNS
BEQ 8$ ;DON'T TEST IF NOT ACTIVE LINE.
;
; CLEAR ALL THE DUT LNCTRL REGISTERS DTR BITS.
;
; CLR R0 ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
; MOV #MAPLNS,R5 ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
; JSR PC,WTWLNC ;CLEAR ALL THE DUT DTR BITS.
```

4652	016666	012704	000050		MOV	#40.,R4	
4653	016672	004767	171244		JSR	PC,DELAY	;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4654					;;		
4655					;;	RECORD THE STATES OF THE MODEM STATUS SIGNALS.	
4656					;;		
4657	016676	004767	172120		JSR	PC,SAVMST	;SAVE THE PRESENT MODEM STATUS STATES.
4658					;;		
4659					;;	SET THE DTR FOR THE SELECTED LINE.	
4660					;;		
4661	016702	010377	163272		MOV	R3,DCSRA	;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
4662	016706	052777	001000	163274	BIS	#BIT9,DLNCTRA	;SET THE SELECTED LINE DTR.
4663	016714	012704	000050		MOV	#40.,R4	
4664	016720	004767	171216		JSR	PC,DELAY	;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
4665					;;		
4666					;;	CHECK THE PRESENT DUT STAT REGISTER CONTENTS AGAINST PREVIOUS.	
4667					;;	IF ANY UNDESIRED CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.	
4668					;;		
4669	016724	116301	003750		MOVB	TXRLNB(R3),R1	;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
4670	016730	012702	120000		MOV	#BIT15:BIT13,R2	;IGNORE DSR AND RI ON ASSOCIATED LINE.
4671	016734	004767	171110		JSR	PC,CMPMST	;COMPARE OLD AND NEW STAT CONTENTS.
4672	016740	103415			BIS	8\$	;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
4673					;;	REPORT INTERACTIONS FOUND BETWEEN DTR FOR LINE NN AND THE FOLLOWING SIGNALS:	
4674	016742	012767	020322	165042	MOV	#8402.,ERRNBR	;SELECT THE ERROR NUMBER.
4675	016750	012767	006720	165040	MOV	#ER8401,ERRBLK	;SELECT THE ERROR PRINT ROUTINE.
4676	016756	012701	005637		MOV	#EM8402,R1	;SELECT THE DTR ERROR MESSAGES.
4677	016762				ERROR		;ER8401 USES R1, R2, AND R3 VALUES.
4678							TRAP C\$ERROR
4679					;;		
4680					;;	EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED	
4681					;;		
4682	016764	032767	000100	163170	BIT	#BIT06,OPTION	;EXIT WITH TEST FAILURE MESSAGE IF
4683	016772	001470			BEQ	60\$	;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4684							;DURING THE SOFTWARE QUESTIONS.
4685					;;		
4686					;;	SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.	
4687					;;		
4688	016774	005203			INC	R3	;SELECT THE NEXT LINE NUMBER.
4689	016776	020327	000020		CMP	R3,#NUMLNS	;TEST FOR ALL LINES DONE.
4690	017002	002716			BLT	6\$	;LOOP IF NOT ALL LINES DONE.
4691					;;		
4692					;;	SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.	
4693					;;	THIS LOOP SETS ALL THE DTRS AND THEN CLEARS THEM INDIVIDUALLY AND CHECKS	
4694					;;	FOR ANY RESPONSES ON SIGNALS OTHER THAN THE ASSOCIATED RI AND DSR SIGNALS.	
4695					;;	THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.	
4696					;;		
4697	017004	005003			CLR	R3	;CLEAR THE LINE COUNTER.
4698	017006	010300			MOV	R3,R0	
4699	017010	006300			ASL	R0	
4700	017012	036067	002310	163152	BIT	BITBL(R0),ACTLNS	
4701	017020	001451			BEQ	12\$	;DON'T TEST IF NOT ACTIVE LINE.
4702					;;		
4703					;;	SET ALL THE DUT LNCTRL REGISTERS DTR BITS.	
4704					;;		
4705	017022	012700	001000		MOV	#BIT9,R0	;SPECIFY THAT DTR BITS ARE TO BE SET.
4706	017026	012705	177777		MOV	#MAPLNS,R5	;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
4707	017032	004767	172372		JSR	PC,WTWLN	;SET ALL THE DUT DTR BITS.

```

4708 017036 012704 000050      MOV    #40.,R4
4709 017042 004767 171074      JSR    PC,DELAY      ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4710                                     ;+
4711                                     ; RECORD THE STATES OF THE MODEM STATUS SIGNALS.
4712                                     ; -
4713 017046 004767 171750      JSR    PC,SAVMST      ;SAVE THE PRESENT MODEM STATUS STATES.
4714                                     ;+
4715                                     ; CLEAR THE DTR FOR THE SELECTED LINE.
4716                                     ; -
4717 017052 010377 163122      MOV    R3,@CSRA      ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
4718 017056 042777 001000 163124 BIC    #BIT9,@LNCTRA ;CLEAR THE SELECTED LINE DTR.
4719 017064 012704 000050      MOV    #40.,R4
4720 017070 004767 171046      JSR    PC,DELAY      ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
4721                                     ;+
4722                                     ; CHECK THE PRESENT DUT STAT REGISTER CONTENTS AGAINST PREVIOUS.
4723                                     ; IF ANY UNDESIRED CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.
4724                                     ; -
4725 017074 116301 003750      MOVB   TXRLNB(R3),R1 ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
4726 017100 012702 120000      MOV    #BIT15:BIT13,R2 ;IGNORE DSR AND RI ON ASSOCIATED LINE.
4727 017104 004767 170740      JSR    PC,CMPMST      ;COMPARE OLD AND NEW STAT CONTENTS.
4728 017110 103415                                     BCS    12$      ;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
4729                                     ;REPORT INTERACTIONS FOUND BETWEEN DTR FOR LINE NN AND THE FOLLOWING SIGNALS:
4730 017112 012767 020323 164672 MOV    #8403.,ERRNBR ;SELECT THE ERROR NUMBER.
4731 017120 012767 006720 164670 MOV    #ER8401,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
4732 017126 012701 005637      MOV    #EM8402,R1 ;SELECT THE DTR ERROR MESSAGES.
4733 017132                                     ERROR ;ER8401 USES R1, R2, AND R3 VALUES.
4734                                     TRAP    C$ERROR
4735
4736                                     ;+
4737                                     ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4738                                     ; -
4738 017134 032767 000100 163020 BIT    #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
4739 017142 001404                                     BEQ    60$      ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4740                                     ;DURING THE SOFTWARE QUESTIONS.
4741
4742                                     ;+
4743                                     ; SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.
4744                                     ; -
4745 017144 005203 000020      12$: INC    R3 ;SELECT THE NEXT LINE NUMBER.
4746 017146 020327 000020      CMP    R3,#NUMLNS ;TEST FOR ALL LINES DONE.
4747 017152 002715                                     BLT    10$      ;LOOP IF NOT ALL LINES DONE.
4748
4749 017154 005067 163042      60$: CLR    CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
4750 017160 012700 000340      SETPRI #PRI07 ;DISABLE ALL INTERRUPTS.
4751                                     MOV    #PRI07,R0
4752 017166                                     TRAP    C$SPRI
4752 017166                                     L10026:
4752 017166 104401                                     TRAP    C$ETST

```

```

.SBTLTTL HARDWARE TEST                                RISINT
;*****
;* - REQUEST TO SEND SIGNAL INTERACTIONS TEST -
;*
;* THIS TEST VERIFIES THAT THE RTS SIGNAL (AND THE LOOPED BACK DCD AND CTS
;* STATUS SIGNALS) DO NOT INTERACT WITH ANY OTHER MODEM STATUS SIGNALS.
;* IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR STAGGERED LOOPBACK IS
;* SPECIFIED. THIS TEST IS PERFORMED ON ALL ACTIVE LINES.
;*****
;--*****

BGNTST

;*****
;* T9::
;* ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
;*-
;* BIT 0BIT1,LOPBCK ;CHECK TYPE OF LOOPBACK MODE SELECTED.
;* BNE 2$
;* JMP 60$ ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
2$: SETPRI 0PRI05 ;ALLOW LTC INTERRUPTS.
;*****
;* MOV 0PRI05,R0
;* TRAP C$SPRI
;* (85)
;* TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
;* MOV 0TNUM,TSTNUM ;SET UP THE TEST NUMBER.
;* MOV 0-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
;* MOV 01,ERRTYP ;SET ERROR TYPE IN ERROR TABLE.
;* MOV 08501,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
;* MOV 0EM8501,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
;*****
;* RESET THE DUT TO A KNOWN STATE. REMOVE STATUS CODES FROM THE FIFO.
;* CLEAR TX AND RX INTERRUPT ENABLE BITS.
;* THIS SUBROUTINE REPORTS ERROR >>>> 8501 <<<<.
;*-
;* JSR PC,CLNRST ;RESET THE DUT.
;* BCS 4$
;* JMP 60$ ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
;*****
;* SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
;*-
4$: JSR PC,ASLNTL ;SET UP THE ASSOCIATED LINE TABLES.
;*****
;* SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
;* THIS LOOP CLEARS ALL THE RTSS AND THEN SETS THEM INDIVIDUALLY AND CHECKS
;* FOR ANY RESPONSES ON SIGNALS OTHER THAN THE ASSOCIATED DCD AND CTS SIGNALS.
;* THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
;*-
6$: CLR R3 ;CLEAR THE LINE COUNTER.
;* MOV R3,R0
;* ASL R0
;* BIT BITTBL(R0),ACTLNS
;* BEQ 8$ ;DON'T TEST IF NOT ACTIVE LINE.
;*****
;* CLEAR ALL THE DUT LNCTRL REGISTERS RTS BITS.
;*-
;* CLR R0 ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
;* MOV 0MAPLNS,R5 ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
;* JSR PC,WTWLNLC ;CLEAR ALL THE DUT RTS BITS.

```

4808	017316	012704	000050		MOV	#40.,R4	
4809	017322	004767	170614		JSR	PC,DELAY	;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4810							
4811							
4812							; RECORD THE STATES OF THE MODEM STATUS SIGNALS.
4813	017326	004767	171470		JSR	PC,SAVMST	;SAVE THE PRESENT MODEM STATUS STATES.
4814							
4815							; SET THE RTS FOR THE SELECTED LINE.
4816							
4817	017332	010377	162642		MOV	R3,0CSRA	;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
4818	017336	052777	010000	162644	BIS	#BIT12,0LNCTRA	;SET THE SELECTED LINE RTS.
4819	017344	012704	000050		MOV	#40.,R4	
4820	017350	004767	170566		JSR	PC,DELAY	;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
4821							
4822							; CHECK THE PRESENT DUT STAT REGISTER CONTENTS AGAINST PREVIOUS.
4823							; IF ANY UNDESIRED CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.
4824							
4825	017354	116301	003750		MOVB	TXRLNB(R3),R1	;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
4826	017360	012702	014000		MOV	#BIT12!BIT11,R2	;IGNORE DCD AND CTS ON ASSOCIATED LINE.
4827	017364	004767	170460		JSR	PC,CMPMST	;COMPARE OLD AND NEW STAT CONTENTS.
4828	017370	103415			BCS	8\$	;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
4829							;REPORT INTERACTIONS FOUND BETWEEN RTS FOR LINE NN AND THE FOLLOWING SIGNALS:
4830	017372	012767	020466	164412	MOV	#8502.,ERRNBR	;SELECT THE ERROR NUMBER.
4831	017400	012767	006720	164410	MOV	#ER8401,ERRBLK	;SELECT THE ERROR PRINT ROUTINE.
4832	017406	012701	005744		MOV	#EM8502,R1	;SELECT THE RTS ERROR MESSAGES.
4833	017412				ERROR		;ER1901 USES R1, R2, AND R3 VALUES.
4834	017412	104460					TRAP C\$ERROR
4835							
4836							
4837							; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4838	017414	032767	000100	162540	BIT	#BIT06,OPTION	;EXIT WITH TEST FAILURE MESSAGE IF
4839	017422	001470			BEQ	60\$	;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4840							;DURING THE SOFTWARE QUESTIONS.
4841							
4842							; SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.
4843							
4844	017424	005203			8\$:	INC R3	;SELECT THE NEXT LINE NUMBER.
4845	017426	020327	000020		CMP	R3,#NUMLNS	;TEST FOR ALL LINES DONE.
4846	017432	002716			BLT	6\$	;LOOP IF NOT ALL LINES DONE.
4847							
4848							; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
4849							; THIS LOOP SETS ALL THE RTSS AND THEN CLEARS THEM INDIVIDUALLY AND CHECKS
4850							; FOR ANY RESPONSES ON SIGNALS OTHER THAN THE ASSOCIATED DCD AND CTS SIGNALS.
4851							; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
4852							
4853	017434	005003			CLR	R3	;CLEAR THE LINE COUNTER.
4854	017436	010300			MOV	R3,R0	
4855	017440	006300			ASL	R0	
4856	017442	036067	002310	162522	BIT	BIT1BL(R0),ACTLNS	
4857	017450	001451			BEQ	12\$	;DON'T TEST IF NOT ACTIVE LINE.
4858							
4859							; SET ALL THE DUT LNCTRL REGISTERS RTS BITS.
4860							
4861	017452	012700	010000		MOV	#BIT12,R0	;SPECIFY THAT RTS BITS ARE TO BE SET.
4862	017456	012705	177777		MOV	#MAPLNS,R5	;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
4863	017462	004767	171742		JSR	PC,WTWLC	;SET ALL THE DUT RTS BITS.

```

4864 017466 012704 000050      MOV    #40.,R4
4865 017472 004767 170444      JSR    PC,DELAY      ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4866                                ;+
4867                                ; RECORD THE STATES OF THE MODEM STATUS SIGNALS.
4868                                ; -
4869 017476 004767 171320      JSR    PC,SAVMST      ;SAVE THE PRESENT MODEM STATUS STATES.
4870                                ;+
4871                                ; CLEAR THE RTS FOR THE SELECTED LINE.
4872                                ; -
4873 017502 010377 162472      MOV    R3,0CSRA      ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
4874 017506 042777 010000      BIC    #BIT12,0LNCTRA ;CLEAR THE SELECTED LINE RTS.
4875 017514 012704 000050      MOV    #40.,R4
4876 017520 004767 170416      JSR    PC,DELAY      ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
4877                                ;+
4878                                ; CHECK THE PRESENT DUT STAT REGISTER CONTENTS AGAINST PREVIOUS.
4879                                ; IF ANY UNDESIRED CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.
4880                                ; -
4881 017524 116301 003750      MOV    TXRLNB(R3),R1 ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
4882 017530 012702 014000      MOV    #BIT12!BIT11,R2 ;IGNORE DCD AND CTS ON ASSOCIATED LINE.
4883 017534 004767 170310      JSR    PC,CMPMST      ;COMPARE OLD AND NEW STAT CONTENTS.
4884 017540 103415              BCS    12$          ;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
4885                                ;REPORT INTERACTIONS FOUND BETWEEN RTS FOR LINE NN AND THE FOLLOWING SIGNALS:
4886 017542 012767 020467      MOV    #8503.,ERRNBR ;SELECT THE ERROR NUMBER.
4887 017550 012767 006720      MOV    #ER8401,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
4888 017556 012701 005744      MOV    #EM8502,R1 ;SELECT THE RTS ERROR MESSAGES.
4889 017562              ERROR      ;ER1901 USES R1, R2, AND R3 VALUES.
4890                                TRAP    C$ERROR
4891                                ;+
4892                                ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4893                                ; -
4894 017564 032767 000100      BIT    #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
4895 017572 001404              BEQ    60$          ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4896                                ;DURING THE SOFTWARE QUESTIONS.
4897                                ;+
4898                                ; SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.
4899                                ; -
4900 017574 005203              12$: INC    R3          ;SELECT THE NEXT LINE NUMBER.
4901 017576 020327 000020      CMP    R3,#NUMLNS ;TEST FOR ALL LINES DONE.
4902 017602 002715              BLT    10$          ;LOOP IF NOT ALL LINES DONE.
4903                                ;+
4904 017604 005067 162412      60$: CLR    CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
4905 017610              SETPRI  #PRI07 ;DISABLE ALL INTERRUPTS.
4906                                MOV    #PRI07,R0
4907 017616              TRAP    C$SPRI
4908 017616              ENDTST
4909 017616 104401              L10027: TRAP    C$ETST

```

```

4909 .SBTTL  HARDWARE TEST          - REP8MP
4910 ;... *****
4911 ;*      REPORT ANY BMP CODES IN THE QUEUE
4912 ;*      THIS IS A PSEUDO-TEST USED TO REPORT ANY BMP CODES THAT WERE FOUND
4913 ;*      IN THE OUT'S FIFO DURING PREVIOUS TEST, AND LOGGED IN THE BMP CODE
4914 ;*      QUEUE.
4915 ;*      IT IS UNLIKELY THAT RUNNING THIS PSEUDO-TEST ALONE WILL PRODUCE ANY
4916 ;*      ERROR REPORTS.
4917 ;*
4918 ;- *****
4919 017620      BGNTST
4920 017620      T10::
4921 017620 000012      TNUM -- TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
4922 017626 012767 000012 162400      MOV #TNUM,TSTNUM      ;SET UP THE TEST NUMBER. (93)
4923 017634 012767 177777 162366      MOV #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
4924 017640 016702 162510      MOV BMPCQP,R2      ;GET THE CONTENTS OF THE POINTER.
4925 017644 012703 002352      MOV #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
4926 017646 020203      CMP R2,R3      ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
4927 017646 001411      BEQ 60$      ;EXIT NO CODES IN THE QUEUE.
4928 ;*
4929 ; THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
4930 ;-
4931 ;REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN"
4932 017650 012701 006124      MOV #EM9304,R1      ;PASS THE FIRST MESSAGE TO BE REORTED.
4933 017654      ERRDF 9301,EM9301,ER9301 ;
4934 017654 104455      >>>> ERROR #9301 <<<<.
4935 017656 022125      TRAP C$ERDF
4936 017660 005750      .WORD 9301
4937 017662 007176      .WORD EM9301
4938 017662 007176      .WORD ER9301
4939 017664 012767 002352 162456      MOV #BMPCQB,BMPCQP ;SET POINTER BACK TO THE BEGINING OF THE QUE.
4940 017672 005067 162324      60$: CLR CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
4941 017676      ENDTST
4942 017676 104401      L10030:
4943 017676 104401      TRAP C$ETST

```

```
4947
4948
4949
4950      .SBTTL  HARDWARE PARAMETER CODING SECTION
4951
4952
4953      ;**
4954      ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
4955      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
4956      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
4957      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
4958      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
4959      ; WITH THE OPERATOR.
4960      ;--
4961
4962      017700      BGNMRD
4963      017700      000016
4964      017702
4965
4966      ;DEVICE CSR ADDRESS QUESTION:
4967      GPRMA  HWPTQ1,0,0,160000,177776,YES
4968
4969      ;ACTIVE LINES BIT MAP QUESTION:
4970      GPRMD  HWPTQ2,2,0,MAPLNS,0,177777,YES
4971
4972      ;TYPE OF LOOPBACK QUESTION:
4973      GPRMD  HWPTQ3,4,0,377,1,3,YES
4974
4975      ENDMRD
4976
4977      L10031:
4978      .EVEN
4979
4980      L10031:
4981      .WORD  L10031-L$HARD/2
4982
4983      .WORD  T$CODE
4984      .WORD  HWPTQ1
4985      .WORD  T$LOLIM
4986      .WORD  T$HILIM
4987
4988      .WORD  T$CODE
4989      .WORD  HWPTQ2
4990      .WORD  MAPLNS
4991      .WORD  T$LOLIM
4992      .WORD  T$HILIM
4993
4994      .WORD  T$CODE
4995      .WORD  HWPTQ3
4996      .WORD  377
4997      .WORD  T$LOLIM
4998      .WORD  T$HILIM
4999
5000      L10031:
5001      .ASCIZ  /CSR ADDRESS: /
5002
5003      L10031:
5004      .ASCIZ  /ACTIVE LINE BIT MAP: /
```

	017776	120	072	040
	020001	000		
4992	020002	124	131	120
	020003	105	040	117
	020010	106	040	114
	020013	117	117	120
	020016	102	101	103
	020021	113	040	050
	020024	061	075	111
	020027	116	124	105
	020032	122	116	101
	020035	114	054	040
	020040	062	075	110
	020043	063	060	062
	020046	071	040	117
	020051	122	040	110
	020054	063	062	067
	020057	067	054	040
	020062	063	075	110
	020065	063	062	065
	020070	051	072	000

4993
4994

HWP1Q3: .ASCIIZ /TYPE OF LOOPBACK (1=INTERNAL, 2=M3029 OR M3277, 3=M325):/

.EVEN

```

5003
5004
5005          .SBTTL  SOFTWARE PARAMETER CODING SECTION
5006
5007          ;**
5008          ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
5009          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
5010          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
5011          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
5012          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
5013          ; WITH THE OPERATOR.
5014          ;
5015
5016          BGNSFT
5017          020074      000014
5018          020074
5019          020076
5020
5021          ;UNIT NUMBER PRINTOUT QUESTION:
5022          GPRML  SWPTQ1,0,20,YES
5023
5024          .WORD  L10032 L$SOFT/2
5025          L$SOFT:
5026
5027          020076      000130
5028          020076      020126
5029          020100      000020
5030          020102
5031
5032          ;EXTENDED ERROR REPORTING QUESTION:
5033          GPRML  SWPTQ2,0,100,YES
5034
5035          .WORD  T$CODE
5036          .WORD  SWPTQ1
5037          .WORD  20
5038
5039          020104      000130
5040          020104      020202
5041          020106      000100
5042          020110
5043
5044          ;**
5045          ; IF EXTENDED ERROR REPORTING IS NOT REQUIRED THEN SKIP THE NEXT QUESTION.
5046          ;**
5047          XFERF  ENDD
5048
5049          .WORD  T$CODE
5050
5051          ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
5052          GPRMD  SWPTQ3,2,D,177777,0,177777,YES
5053
5054          .WORD  T$CODE
5055          .WORD  SWPTQ3
5056          .WORD  177777
5057          .WORD  T$LOLIM
5058          .WORD  T$HILIM
5059
5060          .EVEN
5061
5062          ENDD:  ENDSFT
5063
5064          .EVEN
5065          L10032:
5066
5067          5041  020126      122      105      120
5068          5042  020131      117      122      124
5069          5049  020134      040      125      116
5070          020137      111      124      040
5071          020142      116      125      115
5072          020145      102      105      122
5073          020150      040      101      123
5074          020153      040      105      101
5075          020156      103      110      040

```

	020161	125	116	111
	020164	124	040	111
	020167	123	040	124
	020172	105	123	124
	020175	105	104	072
	020200	040	000	
5050	020202	105	130	124
	020205	105	116	104
	020210	105	104	040
	020213	105	122	122
	020216	117	122	040
	020221	122	105	120
	020224	117	122	124
	020227	111	116	107
	020232	072	040	000
5051	020235	116	125	115
	020240	102	105	122
	020243	040	117	106
	020246	040	111	116
	020251	104	111	126
	020254	111	104	125
	020257	101	114	040
	020262	104	101	124
	020265	101	040	105
	020270	122	122	117
	020273	122	123	040
	020276	124	117	040
	020301	122	105	120
	020304	117	122	124
	020307	040	117	116
	020312	040	101	040
	020315	114	111	116
	020320	105	072	040
5052	020323	000		

SWPTQ2: .ASCIZ /EXTENDED ERROR REPORTING: /

SWPTQ3: .ASCIZ /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /

.EVEN

CZDMWBO DMU 11 FUNC 1ST PART3 MACRO M1200 15 MAR 84 09:41 PAGE 86
SOFTWARE PARAMETER CODING SECTION

```

5061
5062
5063 020324          $PATCH::
5064 020324          .BLKW    24
5065
5072
5073
5074
5075
5076 020374          LASTAD
                    020374 000000          .EVEN
                    020376 000000          .WORD    0
                    020400          .WORD    0
5077 020400          L$LAST::
5078                    ENDMOD
5079
5080
5081
5082
5083
5084
5085          000001          .END

```

ACTLNS	002172	G	C\$AUTO=	000061	EDROP	012512	EVL	=	000004	G	IES	T	002230	G	
ADR	=	000020	G	C\$BRK =	000022	EF.CON=	000036	G	EXOERR	002224	G	ISR	=	000100	G
ADRPTR	010010	G	C\$BSEG=	000004	EF.NEW=	000035	G	E\$END =	002100		IXE	=	004000	G	
ALTFLD	007370	G	C\$BSUB=	000002	EF.PWR=	000034	G	E\$LOAD=	000035		I\$AU	=	000041		
ASLNTL	007442	G	C\$CEFG=	000045	EF.RES=	000037	G	FDATA	002206	G	I\$AUTO=	000041			
ASSEMB=	000010		C\$CLCK=	000062	EF.STA=	000040	G	FDATO =	000006	G	I\$CLN =	000041			
BCOUNT	002274	G	C\$CLEA=	000012	EF0503	004116	G	FSLSA	002206	G	I\$DU =	000041			
BITTBL	002310	G	C\$CLOS=	000035	EF1601	004123	G	FSLSO =	000006	G	I\$HRD =	000041			
BITO	=	000001	G	C\$CLP1=	000006	EF7801	004155	G	F\$AU =	000015		I\$INIT=	000041		
BIT00	=	000001	G	C\$CVEC=	000036	EF8401	004213	G	F\$AUTO=	000020		I\$MOD =	000041		
BIT01	=	000002	G	C\$DCLN=	000044	EF8402	004305	G	F\$BGN =	000040		I\$MSG =	000041		
BIT02	=	000004	G	C\$DDDU=	000051	EF9301	004423	G	F\$CLEA=	000007		I\$PROT=	000040		
BIT03	=	000010	G	C\$DRPT=	000024	EF9302	004501	G	F\$DU =	000016		I\$PTAB=	000041		
BIT04	=	000020	G	C\$DU =	000053	EM0101	010376	G	F\$END =	000041		I\$PWR =	000041		
BIT05	=	000040	G	C\$EDIT=	000003	EM0102	010462	G	F\$HARD=	000004		I\$RPT =	000041		
BIT06	=	000100	G	C\$ERDF=	000055	EM0103	004632	G	F\$HW =	000013		I\$SEG =	000041		
BIT07	=	000200	G	C\$ERHR=	000056	EM1601	004670	G	F\$INIT=	000006		I\$SETU=	000041		
BIT08	=	000400	G	C\$ERRO=	000060	EM7801	004753	G	F\$JMP =	000050		I\$SFT =	000041		
BIT09	=	001000	G	C\$ERSF=	000054	EM7802	005015	G	F\$MOD =	000000		I\$SRV =	000041		
BIT1	=	000002	G	C\$ERSO=	000057	EM7901	005035	G	F\$MSG =	000011		I\$SUB =	000041		
BIT10	=	002000	G	C\$ESCA=	000010	EM7902	005077	G	F\$PROT=	000021		I\$TST =	000041		
BIT11	=	004000	G	C\$ESEG=	000005	EM8001	005117	G	F\$PWR =	000017		J\$JMP =	000167		
BIT12	=	010000	G	C\$ESUB=	000003	EM8002	005163	G	F\$RPT =	000012		LGRP1M	002232	G	
BIT13	=	020000	G	C\$ETST=	000001	EM8101	005227	G	F\$SEG =	000003		LGRP2M	002234	G	
BIT14	=	040000	G	C\$EXIT=	000032	EM8102	005272	G	F\$SOFT=	000005		LNCTRA	002210	G	
BIT15	=	100000	G	C\$GETB=	000026	EM8201	005335	G	F\$SRV =	000010		LNCTRO=	000010	G	
BIT2	=	000004	G	C\$GETW=	000027	EM8202	005401	G	F\$SUB =	000002		LOE	=	040000	G
BIT3	=	000010	G	C\$GMAN=	000043	EM8301	005445	G	F\$SW =	000014		LOPBCK	002174	G	
BIT4	=	000020	G	C\$GPHR=	000042	EM8302	005511	G	F\$TEST=	000001		LOT	=	000010	G
BIT5	=	000040	G	C\$GPLO=	000030	EM8401	005555	G	GETPRM	012170		LPCSLT=	000036	G	
BIT6	=	000100	G	C\$GPRI=	000040	EM8402	005637	G	G\$CNTO=	000200		LPRA	002204	G	
BIT7	=	000200	G	C\$INIT=	000011	EM8403	005643	G	G\$DELM=	000372		LPRO	=	000004	G
BIT8	=	000400	G	C\$INLP=	000020	EM8404	005647	G	G\$DISP=	000003		L\$ACP	002110	G	
BIT9	=	001000	G	C\$MANI=	000050	EM8405	005652	G	G\$EXCP=	000400		L\$APT	002036	G	
BMPCQB	002352	G	C\$MEM =	000031	EM8406	005656	G	G\$HILI=	000002		L\$AU	012520	G		
BMPCQE	002552	G	C\$MSG =	000023	EM8501	005662	G	G\$LOLI=	000001		L\$AUT	002070	G		
BMPCQP	002350	G	C\$OPEN=	000034	EM8502	005744	G	G\$NO =	000000		L\$AUTO	012370	G		
BOE	=	000400	G	C\$PNTB=	000014	EM9301	005750	G	G\$OFFS=	000400		L\$CCP	002106	G	
BRLEVL	002175	G	C\$PNTF=	000017	EM9302	006027	G	G\$OFSI=	000376		L\$CLEA	012372	G		
BUFBAS	002650	G	C\$PNTS=	000016	EM9303	006057	G	G\$PRMA=	000001		L\$CO	002032	G		
BUFEND	003650	G	C\$PNTX=	000015	EM9304	006124	G	G\$PRMD=	000002		L\$DEPO	002011	G		
BUFMID	003250	G	C\$QIO =	000377	END0	020126		G\$PRML=	000000		L\$DESC	004070	G		
BUFPTR	002220	G	C\$RDBU=	000007	ENDET8	003650	G	G\$RADA=	000140		L\$DESP	002076	G		
BUF3QT	003450	G	C\$REFG=	000047	ENDIT	012354		G\$RADB=	000000		L\$DEVP	002060	G		
CALMSL	007552	G	C\$RESE=	000033	ERCNT8	002554	G	G\$RADL=	000120		L\$DISP	002124	G		
CKTRAP	007776	G	C\$REVI=	000003	ERLTBL	002650	G	G\$RADO=	000020		L\$DLY	002116	G		
CLKBRL	002260	G	C\$RFLA=	000021	ERRBLK	004016	G	G\$XFER=	000004		L\$DTP	002040	G		
CLKCSR	002256	G	C\$RPT =	000025	ERRMSG	004014	G	G\$YES =	000010		L\$DTYP	002034	G		
CLKHRZ	002264	G	C\$SEFG=	000046	ERRNBR	004012	G	HELP	=	000000	L\$DU	012410	G		
CLKINT	011460	G	C\$SPRI=	000041	ERRTYP	004010	G	HOE	=	100000	G	L\$DUT	002072	G	
CLKVEC	002262	G	C\$SVEC=	000037	ERSMRF	002552	G	HWPTQ1	017736		L\$DVTY	004060	G		
CLNRST	010026	G	C\$TPRI=	000013	ER0101	006200	G	HWPTQ2	017754		L\$EF	002052	G		
CMPMST	010050	G	DELAY	010142	G	ER0503	006532	G	HWPTQ3	020002		L\$ENVI	002044	G	
CSRA	002200	G	DFPTBL	002152	G	ER1603	006570	G	IBE	=	010000	G	L\$ERRT	004010	G
CSRO	=	000000	G	DIAGMC=	000000	ER7801	006662	G	IDU	=	000040	G	L\$ETP	002102	G
CTRLCF	002222	G	DRADRT	002200	G	ER8401	006720	G	IER	=	020000	G	L\$EXP1	002046	G
C\$AU	=	000052		DROP	012434	ER9301	007176	G				L\$EXP4	002064	G	

L\$XP5	002066 G	L10017	013006	PNT	= 001000 G	SWPTQ1	020126	T\$SUBN=	000000
L\$HARD	017702 G	L10020	013542	PREGRT	004042 G	SWPTQ2	020202	T\$TAGL=	177777
L\$HIME	002120 G	L10021	014276	PREG05	004020	SWPTQ3	020235	T\$TAGN=	010033
L\$HPCP	002016 G	L10022	014746	PRI	= 002000 G	S\$LSYM=	010000	T\$TEMP=	000000
L\$HPTP	002022 G	L10023	015416	PRI00	= 000000 G	TIMER1	002266 G	T\$TEST=	000012
L\$HW	002152 G	L10024	016066	PRI01	= 000040 G	TIMER2	002270 G	T\$TSTM=	177777
L\$ICP	002104 G	L10025	016536	PRI02	= 000100 G	TIMER3	002272 G	T\$TSTS=	000001
L\$INIT	011566 G	L10026	017166	PRI03	= 000140 G	TNUM	= 000012 G	T\$AU	= 010016
L\$LADP	002026 G	L10027	017616	PRI04	= 000200 G	TP4FLG	002252 G	T\$AUT=	010013
L\$LAST	020400 G	L10030	017676	PRI05	= 000240 G	TP4RTN	011530 G	T\$CLE=	010014
L\$LOAD	002100 G	L10031	017736	PRI06	= 000300 G	TP4VEC	002250 G	T\$DU	= 010015
L\$LUN	002074 G	L10032	020126	PRI07	= 000340 G	TSTNUM	002226 G	T\$HAR=	010031
L\$MREV	002050 G	MAPLNS=	177777 G	PUFIFO	010560 G	TXAD1A	002212 G	T\$HL=	010000
L\$NAME	002000 G	MFUNIT	004601 G	RBUFA	002202 G	TXAD10=	000012 G	T\$INI=	010012
L\$PRIO	002042 G	MMENAB	002306 G	RBUFO	= 000002 G	TXAD2A	002214 G	T\$MSG=	010007
L\$PROT	011560 G	MMPRES	002304 G	RESETT	010642 G	TXAD20=	000014 G	T\$PRO=	010011
L\$PRT	002112 G	MMSRO	002302 G	RXBCTX=	000030 G	TXBFCA	002216 G	T\$RPT=	010010
L\$REPP	002062 G	MSG1	006316 G	RXBCTX=	000020 G	TXBFCO=	000016 G	T\$SOF=	010032
L\$REV	002010 G	MSG2	006374 G	RXBFUL=	000100 G	TXJNTC	002244 G	T\$SW	= 010001
L\$RPT	011552 G	MSG3	006453 G	RXINTC	002240 G	TXINTF	002246 G	T\$TES=	010030
L\$SOFT	020076 G	MSLCNT	002300 G	RXINTF	002242 G	TXRLNB	003750 G	T1	012526 G
L\$SPC	002056 G	MSLGET	010202 G	RXTIMO=	000002 G	TXRLNE	003770 G	T10	017620 G
L\$SPCP	002020 G	MSLOOP	010316 G	RXTMA	002202 G	TXRXLB	003710 G	T2	013010 G
L\$STP	002024 G	MSTICK	002276 G	RXVECA	002166 G	TXRXLE	003750 G	T3	013544 G
L\$STA	002030 G	NDERPT	002164 G	R0SLOT=	000002 G	TXVECA	002170 G	T4	014300 G
L\$SW	002162 G	NEWPAS	012150	R1SLOT=	000004 G	T\$ARGC=	000002	T5	014750 G
L\$TEST	002114 G	NEWRES	012142	R2SLOT=	000006 G	T\$CODE=	001052	T6	015420 G
L\$TIML	002014 G	NEWSTA	011632	R3SLOT=	000010 G	T\$ERRN=	022125	T7	016070 G
L\$UNIT	002012 G	NUMLNS=	000020 G	R4SLOT=	000012 G	T\$EXCP=	000000	T8	016540 G
L10000	002160	OOPS	010332 G	R5SLOT=	000014 G	T\$FLAG=	000050	T9	017170 G
L10001	002166	OPTION	002162 G	SAVBMP	010754 G	T\$GMAN=	000000	UAM	= 000200 G
L10002	006314	O\$APTS=	000000	SAVMST	011022 G	T\$HILI=	177777	UNITN	002176 G
L10003	006566	O\$AU	= 000000	SFPTBL	002162 G	T\$LAST=	000001	UNSDIV	011144 G
L10004	006660	O\$BGNR=	000001	SKPSTS	011066 G	T\$LOLI=	000000	WAIBIC	011300 G
L10005	006716	O\$BGNS=	000001	STGTRB	003770 G	T\$LSYM=	010000	WAIBIS	011354 G
L10006	007174	O\$DU	= 000001	STSTB	002610 G	T\$LTNO=	000012	WORD1	002254 G
L10007	007366	O\$ERRT=	000001	STSTE	002650 G	T\$NEST=	177777	WTWLC	011430 G
L10010	011556	O\$GNSW=	000001	SVCGBL=	000000	T\$NSO	= 000000	X\$ALWA=	000000
L10012	012366	O\$POIN=	000001	SVCINS=	000001	T\$NS1	= 000005	X\$FALS=	000040
L10013	012370	O\$SETU=	000000	SVCSUB=	000001	T\$PTNU=	000000	X\$OFFS=	000400
L10014	012406	PASCNT	002236 G	SVCTAG=	000001	T\$SAVL=	177777	X\$TRUE=	000020
L10015	012516	PCSL0T=	000016 G	SVCTST=	000001	T\$SEGL=	177777	\$PATCH	020324 G
L10016	012524								

. ABS. 020400 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 28661 WORDS (112 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:02:41
CZDHMB0.BIN,CZDHMB0.LST/-SP=SVC34R/ML,CZDHMB0.P11